

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

На правах рукописи

Исаченко Роман Владимирович

СНИЖЕНИЕ РАЗМЕРНОСТИ ПРОСТРАНСТВА
В ЗАДАЧАХ ДЕКОДИРОВАНИЯ СИГНАЛОВ

05.13.17 — Теоретические основы информатики

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н. В. В. Стрижов

Москва — 2021

Оглавление

	Стр.
Введение	4
Глава 1. Постановка задачи декодирования сигналов	13
1.1 Регрессионная модель в пространстве высокой размерности	13
1.2 Задача декодирования сигналов	15
1.3 Обзор методов снижения размерности для задачи декодирования . .	19
Глава 2. Задача построения согласованных моделей декодирования	26
2.1 Процесс согласования моделей в пространстве высокой размерности	26
2.2 Доказательство корректности алгоритма проекции в скрытое про- странство	30
2.3 Аддитивная суперпозиция моделей декодирования	34
2.4 Анализ линейных методов проекции в скрытое пространство	38
2.5 Анализ нелинейных методов проекции в скрытое пространство . . .	40
Глава 3. Выбор признаков в задаче декодирования сигналов	46
3.1 Выбор признаков с помощью квадратичного программирования . . .	47
3.2 Методы выбора признаков для случая векторной целевой переменной	48
3.3 Анализ методов учета значимостей целевых переменных	56
Глава 4. Выбор параметров нелинейных моделей с помощью квадратичного отбора признаков	62
4.1 Задача выбора параметров для оптимизации нелинейных моделей . .	63
4.2 Метод Ньютона для оптимизации параметров	64
4.3 Метод Ньютона с выбором параметров с помощью квадратичного программирования	67
4.4 Анализ значимостей параметров нелинейных моделей	69

Глава 5. Метрические методы анализа временных рядов	74
5.1 Метрическое обучение в задачах кластеризации временных рядов	74
5.2 Алгоритм адаптивного метрического обучения	75
5.3 Задача метрического обучения с динамическим выравниванием временных рядов	78
5.4 Анализ метрического пространства для задачи кластеризации	82
5.5 Анализ метрического пространства для задачи классификации временных рядов	85
Глава 6. Порождение признаков с помощью метамоделей	91
6.1 Постановка задачи порождения признакового пространства	91
6.2 Модели порождения признакового пространства для временных рядов	92
6.3 Классификация временных рядов в порожденном признаковом пространстве	95
6.4 Анализ порожденных признаковых пространств	97
Заключение	103
Список основных обозначений	105
Список иллюстраций	106
Список таблиц	109
Список литературы	111

Введение

Актуальность темы. В работе исследуется проблема снижения размерности пространства при решении задачи декодирования сигналов. Процесс декодирования заключается в восстановлении зависимости между двумя гетерогенными наборами данных. Прогностическая модель предсказывает набор целевых сигналов по набору исходных сигналов.

Исходное описание данных является избыточным. При высокой мультикорреляции в исходном и целевом пространствах финальная прогностическая модель оказывается неустойчивой. Для построения простой, устойчивой и точной модели применяются методы снижения размерности пространства [1, 2, 3, 4] и выбора признаков [5, 6].

В работе решается задача декодирования с векторной целевой переменной. Пространство целевых сигналов содержит скрытые зависимости. Методы снижения размерности, не учитывающие зависимости в целевом пространстве, не являются адекватными. При предсказании векторной целевой переменной анализируется структура целевого пространства. Предложены методы, которые учитывают зависимости как в пространстве исходных сигналов, так и в пространстве целевых сигналов. Предлагается отобразить пространства исходных и целевых сигналов в скрытые подпространства меньшей размерности. Для построения оптимальной модели предлагаются методы согласования скрытых пространств [7, 8, 9]. Предложенные методы позволяют учесть регрессионную компоненту между исходным и целевым сигналами, а также авторегрессионную компоненту целевого сигнала.

Методы снижения размерности пространства понижают размерность исходного пространства, и, как следствие, сложность модели существенно снижается [10, 11, 7, 12]. Алгоритмы снижения размерности находят оптимальные комбинации исходных признаков. Если число таких комбинаций существенно меньше, чем число исходных признаков, то полученное представление снижает

размерность. Цель снижения размерности — получение наиболее репрезентативных и информативных комбинаций признаков для решения задачи.

Выбор признаков является частным случаем снижения размерности пространства [5, 13]. Найденные комбинации признаков являются подмножеством исходных признаков. Таким образом отсеиваются шумовые неинформативные признаки. Рассматриваются два типа методов выбора признаков [6, 14, 15]. Первый тип методов не зависит от последующей прогностической модели. Признаки отбираются на основе свойств исходных пространств, а не на основе свойств модели. Второй тип методов отбирает признаки с учётом знания о прогностической модели.

После нахождения оптимального представления данных с помощью снижения размерности, ставится задача нахождения оптимальной метрики в скрытом пространстве [16, 17, 18, 19, 20]. В случае евклидова пространства естественным выбором метрики оказывается квадратичная норма. Задача метрического обучения заключается в нахождении оптимальной метрики в скрытом пространстве сигналов.

В качестве прикладной задачи анализируется задача построения нейрокомпьютерного интерфейса [21, 22]. Цель состоит в извлечении информации из сигналов мозговой активности [23, 24, 25]. В качестве исходных сигналов выступают сигналы электроэнцефалограммы или электрокортикограммы. Целевым сигналом является траектория движения конечности индивидуума. Необходимо построить адекватную и эффективную модель декодирования исходного сигнала в целевой сигнал. Пространство частотных характеристик мозговых сигналов и авторегрессионное пространство целевых сигналов являются чрезвычайно избыточными [26, 27]. Построение модели без учёта имеющихся зависимостей приводит к неустойчивости модели.

В диссертации решается задача декодирования с векторной целевой переменной. Для построения оптимальной модели декодирования сигналов предлагаются методы выбора согласованных моделей с проекцией в скрытое простран-

ство. Исходные и целевые сигналы проецируются в пространство существенно меньшей размерности. Для связи проекций исходного и целевого сигналов предлагаются методы согласования. Рассматриваются гетерогенные наборы сигналов, природа источников измерений различны. Рассматриваются как линейные методы декодирования, так и их нелинейные обобщения. Доказаны теоремы об оптимальности предложенных методов выбора моделей.

Цели работы.

1. Исследовать свойства решения задачи декодирования сигналов с векторной целевой переменной.
2. Предложить методы снижения размерности пространства, учитывающие зависимости как в пространстве исходных сигналов, так и в целевом пространстве.
3. Предложить процедуру выбора признаков для задачи декодирования сигналов.
4. Исследовать свойства линейных и нелинейных моделей для решения поставленной модели. Получить теоретические оценки оптимальности моделей.
5. Провести вычислительные эксперименты для проверки адекватности предложенных методов.

Основные положения, выносимые на защиту.

1. Исследована проблема снижения размерности сигналов в коррелированных пространствах высокой размерности. Предложены методы декодирования сигналов, учитывающие зависимости как в исходном, так и в целевом пространстве сигналов.
2. Доказаны теоремы об оптимальности предлагаемых методов декодирования сигналов. Предлагаемые методы выбирают согласованные модели в случае избыточной размерности описания данных.

3. Предложены методы выбора признаков, учитывающие зависимости как в исходном, так и в целевом пространстве. Предложенные методы доставляют устойчивые и адекватные решения в пространствах высокой размерности.
4. Предложены нелинейные методы согласования скрытых пространств. Предложен метод выбора активных параметров для оптимизации нелинейной модели. Исследованы свойства предлагаемого метода.
5. Предложен алгоритм метрического обучения для временных рядов с процедурой их выравнивания.
6. Предложен ряд моделей для прогнозирования гетерогенных наборов сигналов для задачи построения нейрокомпьютерных интерфейсов. Проведены вычислительные эксперименты, подтверждающие адекватность моделей.

Методы исследования. Для достижения поставленных целей используются линейные и нелинейные методы регрессионного анализа. Для анализа временных рядов используются авторегрессионные методы. Для извлечения признаков используются частотные характеристики временного ряда. Для построения скрытого пространства используются линейные методы снижения размерности пространства, их нелинейные модификации, а также нейросетевые методы. Для выбора признаков используются методы, основанные на решении задачи квадратичного программирования. Для построения метрического пространства используются методы условной выпуклой оптимизации.

Научная новизна. Предложены методы построения моделей декодирования сигналов, учитывающие структуры пространств исходных и целевых переменных. Предложены методы проекции сигналов в скрытое пространство, а также процедуры согласования образов. Предложены методы выбора признаков с помощью квадратичного программирования. Предложен метод выбора

активных параметров нелинейной модели с помощью выбора признаков. Предложены методы построения оптимального метрического пространства для задачи анализа временных рядов.

Теоретическая значимость. Доказаны теоремы об оптимальности предлагаемых согласованных моделей декодирования сигналов. Доказаны теоремы о корректности рассматриваемых методов проекций в скрытое пространство. Доказаны теоремы о достижении точки равновесия для предлагаемых методов выбора признаков.

Практическая значимость. Предложенные в работе методы предназначены для декодирования набора временных рядов сигналов электрокортикограмм; выбора оптимальных частотных характеристик сигналов; выбора активных параметров модели; классификации и кластеризации временных рядов физической активности.

Степень достоверности и апробация работы. Достоверность результатов подтверждена математическими доказательствами, экспериментальной проверкой результатов предлагаемых методов на реальных данных, публикациями результатов в рецензируемых научных изданиях, в том числе рекомендованных ВАК. Результаты работы докладывались и обсуждались на следующих научных конференциях.

1. Р. В. Исаченко. Метрическое обучение в задачах мультиклассовой классификации временных рядов. *Международная научная конференция «Ломоносов»*, 2016, [28].
2. R. V. Isachenko, et al. Multimodal forecasting multiscale time series in internet of things. *Международная научная конференция «11th International Conference on Intelligent Data Processing: Theory and Applications»*, 2016, [29].

3. Р. В. Исаченко, и др. Локальные модели для классификации объектов сложной структуры. *Всероссийская научная конференция «Математические методы распознавания образов»*, 2017, [30].
4. R. V. Isachenko. Dimensionality reduction for multicorrelated signal decoding with projections to latent space. *Международная научная конференция «12th International Conference on Intelligent Data Processing: Theory and Applications»*, 2018, [31].
5. Р. В. Исаченко. Снижение размерности в задаче декодирования временных рядов. *Международная научная конференция «13th International Conference on Intelligent Data Processing: Theory and Applications»*, 2020, [32].

Работа поддержана грантами Российского фонда фундаментальных исследований.

1. 19-07-00885, Российский фонд фундаментальных исследований в рамках гранта «Выбор моделей в задачах декодирования временных рядов высокой размерности».
2. 16-37-00485, Российский фонд фундаментальных исследований в рамках гранта «Развитие методов выбора признаков в условиях мультиколлинеарности».
3. 16-07-01160, Российский фонд фундаментальных исследований в рамках гранта «Развитие теории обучения по предпочтениям с использованием частично упорядоченных множеств экспертных оценок».
4. 16-07-01154, Российский фонд фундаментальных исследований в рамках гранта «Новые методы прогнозирования на базе субквадратичного анализа метрических конфигураций».

Публикации по теме диссертации. Основные результаты по теме диссертации изложены в 6 печатных изданиях, 5 из которых изданы в журналах, рекомендованных ВАК.

1. Исаченко Р. В., Катруца А. М. Метрическое обучение и снижение размерности пространства в задачах кластеризации // Машинное обучение и анализ данных, 2016. Т. 2. № 1. С. 17–25 [33].
2. Исаченко Р. В., Стрижов В. В. Метрическое обучение в задачах мультиклассовой классификации временных рядов // Информатика и её применения, 2016. Т. 10. № 2. С. 48–57 [34].
3. Isachenko R. et al. Feature Generation for Physical Activity Classification // Artificial Intelligence and Decision Making, 2018. № 3. С. 20–27 [35].
4. Isachenko R., Strijov V. Quadratic programming optimization with feature selection for nonlinear models // Lobachevskii Journal of Mathematics, 2018. Т. 39. № 9. С. 1179–1187 [36].
5. Isachenko R., Vladimirova M., Strijov V. Dimensionality Reduction for Time Series Decoding and Forecasting Problems // DEStech Transactions on Computer Science and Engineering, 2018. №. optim [37].
6. Исаченко Р.В., Яушев Ф.Ю., Стрижов В.В. Модели согласования скрытого пространства в задаче прогнозирования // Системы и средства информатики, 2021. Т. 31. № 1 [38].

Структура и объем работы. Диссертация состоит из оглавления, введения, 6 глав, заключения, списка иллюстраций, списка таблиц, списка основных обозначений и списка литературы из 112 наименований. Основной текст занимает 122 страниц.

Личный вклад. Все приведенные результаты, кроме отдельно оговоренных случаев, получены диссертантом лично при научном руководстве д.ф.-м.н. В. В. Стрижова.

Краткое содержание работы по главам. В главе 1 вводятся основные понятия и обозначения. В разделе 1.1 формулируется задача восстановления ре-

грессионной зависимости в пространствах высокой размерности. В разделе 1.2 ставится задача декодирования сигналов, приводится обзор методов анализа временных рядов. В разделе 1.3 приводится обзор методов снижения размерности пространства для задачи декодирования сигналов.

Глава 2 посвящена задаче построения согласованной модели декодирования. В разделе 2.1 вводятся понятия скрытого пространства и процесса согласования зависимостей, рассматриваются конкретные примеры методов снижения размерности пространства в терминах задачи согласования проекций. В разделе 2.2 приводится доказательство корректности работы линейных методов проекции в скрытое пространство. Раздел 2.3 посвящен рассмотрению случая аддитивной суперпозиции моделей декодирования, анализируются свойства моделей, входящих в суперпозицию. Раздел 2.4 содержит вычислительный эксперимент, демонстрирующий эффективность рассматриваемых линейных согласованных моделей декодирования сигналов. В разделе 2.5 приводится вычислительный эксперимент для нелинейных модификаций согласованных моделей декодирования.

Глава 3 посвящена методам выбора признаков для задачи декодирования сигналов. Ставится задача выбора признаков как задача минимизации функции ошибки. В разделе 3.1 рассматривается метод выбора признаков с помощью квадратичного программирования для случая скалярной целевой переменной. Раздел 3.2 посвящен обобщению скалярного случая на случай векторной целевой переменной. Приводятся методы выбора признаков, учитывающие зависимости в целевом пространстве. Раздел 3.3 содержит вычислительный эксперимент, показывающий, что предложенные методы доставляют адекватные и устойчивые решения в сильно скоррелированных пространствах.

В главе 4 рассматривается задача выбора активных параметров для оптимизации нелинейных моделей. В разделе 4.1 ставится формальная задача выбора параметров модели как задача минимизации функции ошибки. В разделе 4.2 описан метод Ньютона для задачи нелинейной регрессии с квадратичной функ-

цией потерь, а также для задачи логистической регрессии с кросс-энтропийной функцией потерь. В разделе 4.3 приводится метод выбора активных параметров для рассматриваемых задач, использующий метод выбора признаков с помощью квадратичного программирования. Раздел 4.4 содержит вычислительный эксперимент, доказывающий эффективность выбора параметров на множестве задач.

Глава 5 посвящена построению оптимального метрического пространства для анализа временных рядов. Рассматриваются задачи кластеризации и классификации множества временных рядов сигналов активности человека. В разделе 5.1 ставится задача поиска оптимальной метрики Махalanобиса для задачи кластеризации временных рядов. В разделе 5.2 приводится алгоритм адаптивного метрического обучения для нахождения оптимального метрического пространства. В разделе 5.3 рассматривается задача классификации временных рядов, использующая процедуру динамического выравнивания. Разделы 5.4 и 5.5 содержат вычислительные эксперименты на реальных временных рядах с акселерометра мобильного телефона.

Глава 6 посвящена методам построения оптимального признакового пространства для задачи анализа сигналов. В разделе 6.1 ставится формальная задача порождения признакового описания. Раздел 6.2 содержит описание моделей порождения признакового пространства, основанных на экспертных знаниях и на порождающих моделях временных рядов. В разделе 6.3 рассматривается задача классификации временных рядов по полученным признаковым описаниям. В разделе 6.4 приводится вычислительный эксперимент, сравнивающий различные порождающие модели.

Глава 1

Постановка задачи декодирования сигналов

В данной главе ставится общая задача декодирования временных рядов. Приводится обзор стандартных методов анализа временных рядов. Ставится задача построения оптимальной линейной регрессионной модели декодирования. Приведен обзор методов снижения размерности пространства, их обобщений и модификаций.

1.1 Регрессионная модель в пространстве высокой размерности

Пусть $\mathbb{X} \subset \mathbb{R}^n$ — пространство исходной переменной, $\mathbb{Y} \subset \mathbb{R}^r$ — пространство целевой переменной. Пусть задано множество пар $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, где $\mathbf{x}_i \in \mathbb{X}$ — вектор исходной переменной, $\mathbf{y}_i \in \mathbb{Y}$ — вектор целевой переменной.

Обозначим за $\mathbf{X} \in \mathbb{R}^{m \times n}$ исходную матрицу, за $\mathbf{Y} \in \mathbb{R}^{n \times k}$ целевую матрицу:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^{\top} = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^{\top} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r].$$

Столбцы $\{\boldsymbol{\chi}_j\}_{j=1}^n$ матрицы \mathbf{X} являются исходными признаками, столбцы $\{\boldsymbol{\nu}_j\}_{j=1}^r$ матрицы \mathbf{Y} являются целевыми столбцами.

Предполагается, что между исходной переменной \mathbf{x} и целевой переменной \mathbf{y} существует зависимость. Требуется построить прогностическую модель $\mathbf{f} : \mathbb{X} \rightarrow \mathbb{Y}$ из пространства исходной переменной в пространство целевой переменной.

Задача восстановления регрессионной зависимости состоит в нахождении оптимальной модели \mathbf{f}^* по заданным матрицам \mathbf{X} и \mathbf{Y} . Под оптимальностью понимается нахождение такой модели, которая бы доставляла минимум некоторой функции ошибки \mathcal{L} :

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} \mathcal{L}(\mathbf{f}, \mathbf{X}, \mathbf{Y}). \quad (1.1)$$

Задача поиска оптимальной модели является задачей функциональной оптимизации. Для сужения пространства поиска моделей будем рассматривать

параметрические модели $\mathbf{f}(\mathbf{x}, \Theta)$, где Θ — параметры модели. Таким образом между векторами \mathbf{x} и \mathbf{y} существует зависимость вида

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \Theta) + \boldsymbol{\varepsilon},$$

где \mathbf{f} — параметрическая прогностическая модель, Θ — параметры модели, $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ — вектор регрессионных остатков.

Задача (1.1) сводится к задаче поиска оптимальных параметров

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y}). \quad (1.2)$$

В диссертации рассматривается случай избыточной размерности пространств \mathbb{X}, \mathbb{Y} . В таком случае решение задачи (1.2) оказывается неустойчивым. Рассмотрим в качестве примера задачу восстановления линейной регрессии.

Предположим, что зависимость $\mathbf{f}(\mathbf{x}, \Theta)$ линейная:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \Theta) + \boldsymbol{\varepsilon} = \Theta^\top \mathbf{x} + \boldsymbol{\varepsilon}, \quad (1.3)$$

где $\Theta \in \mathbb{R}^{n \times r}$ — матрица параметров модели.

Оптимальные параметры Θ определяются минимизацией функции ошибки $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$. При решении задачи линейной регрессии в качестве такой функции ошибки рассматривается квадратичная функция потерь:

$$\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y}) = \left\| \mathbf{Y}_{m \times r} - \mathbf{X}_{m \times n} \cdot \Theta_{r \times n} \right\|_2^2 \rightarrow \min_{\Theta}. \quad (1.4)$$

Решением (1.4) является следующая матрица:

$$\Theta = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

Наличие линейной зависимости между столбцами матрицы \mathbf{X} приводит к неустойчивому решению задачи оптимизации (1.4). Если существует вектор $\boldsymbol{\alpha} \neq \mathbf{0}_n$ такой, что $\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}_m$, то добавление $\boldsymbol{\alpha}$ к любому столбцу матрицы Θ не меняет значение функции потерь $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$. В этом случае матрица $\mathbf{X}^\top \mathbf{X}$

близка к сингулярной и не обратима. Чтобы избежать сильной линейной зависимости между признаками, в данной работе исследуются методы снижения размерности и выбора признаков.

1.2 Задача декодирования сигналов

Задача декодирования сигналов состоит в восстановлении регрессионной зависимости (1.1) между наборами гетерогенных сигналов.

Пусть имеется два множества временных рядов $\mathcal{S}_x = \{\mathbf{s}_x^i\}_{i=1}^m$ и $\mathcal{S}_y = \{\mathbf{s}_y^i\}_{i=1}^r$, состоящие из m и r временных рядов соответственно. Первое множество \mathcal{S}_x является множеством временных рядов m исходных сигналов. Второе множество \mathcal{S}_y является множеством временных рядов r целевых сигналов. Каждый временной ряд $\mathbf{s} = (s_1, s_2, \dots, s_T)$ является последовательностью измерений некоторой величины в течение времени.

Определение 1. Временное представление $\mathbf{x}_t = ([\mathbf{s}_x^1]_t, \dots, [\mathbf{s}_x^m]_t) \in \mathbb{R}^m$ состоит из измерений временных рядов исходных сигналов в момент времени t . Аналогично временное представление $\mathbf{y}_t = ([\mathbf{s}_y^1]_t, \dots, [\mathbf{s}_y^r]_t) \in \mathbb{R}^r$ состоит из измерений временных рядов целевых сигналов в момент времени t .

Определение 2. Определим представление предыстории длины h для момента времени t множества временных рядов исходных сигналов \mathcal{S}_x как совокупность представлений $\mathbf{X}_{t,h} = [\mathbf{x}_{t-h+1}, \dots, \mathbf{x}_t]^T \in \mathbb{R}^{h \times m}$. Аналогично определим представление предыстории длины h для момента времени t множества временных рядов целевых сигналов \mathcal{S}_y как совокупность представлений $\mathbf{Y}_{t,h} = [\mathbf{y}_{t-h+1}, \dots, \mathbf{y}_t]^T \in \mathbb{R}^{h \times r}$.

Определение 3. Определим представление горизонта прогнозирования длины p для момента времени t множества временных рядов исходных сигналов \mathcal{S}_x как совокупность представлений $\mathbf{X}_{t,p} = [\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+p}]^T \in \mathbb{R}^{p \times m}$. Аналогично определим представление горизонта прогнозирования длины p для момента времени t множества временных рядов целевых сигналов \mathcal{S}_y как совокупность представлений $\mathbf{Y}_{t,p} = [\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+p}]^T \in \mathbb{R}^{p \times r}$.

Задача авторегрессионного декодирования состоит в построении прогностической модели \mathbf{f}^{AR} , дающей прогноз представления горизонта прогнозирования множества временных рядов по представлению предыстории прогнозирования того же множества временных рядов.

Определение 4. Прогностическая модель $\mathbf{f}_{\mathbf{x}}^{\text{AR}} : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{p \times m}$ является *авторегрессионной моделью*, которая по представлению предыстории $\mathbf{X}_{t,h}$ множества временных рядов исходных сигналов $\mathcal{S}_{\mathbf{x}}$ предсказывает представление горизонта прогнозирования $\mathbf{X}_{t,p}$ множества временных рядов исходных сигналов $\mathcal{S}_{\mathbf{x}}$. Аналогично вводится прогностическая модель $\mathbf{f}_{\mathbf{y}}^{\text{AR}} : \mathbb{R}^{h \times r} \rightarrow \mathbb{R}^{p \times r}$ для множества целевых сигналов $\mathcal{S}_{\mathbf{y}}$.

Суть авторегрессионного декодирования заключается в предсказании будущего прогноза сигнала по его же предыстории.

Определение 5. Определим задачу *регрессионного декодирования* как задачу построения прогностической модели $\mathbf{f}_{\mathbf{xy}}^{\text{R}} : \mathbb{R}^{h \times m} \rightarrow \mathbb{R}^{p \times r}$, которая по представлению предыстории $\mathbf{X}_{t,h}$ множества временных рядов исходных сигналов $\mathcal{S}_{\mathbf{x}}$ предсказывает представление горизонта прогнозирования $\mathbf{Y}_{t,p}$ множества временных рядов целевых сигналов $\mathcal{S}_{\mathbf{y}}$.

Отличие регрессионного декодирования от авторегрессионного декодирования состоит в том, что в случае регрессионного декодирования представление предыстории и представление горизонта прогнозирования получены из временных рядов разных пространств. Предыстория получена из множества исходных сигналов, в то время как горизонт прогнозирования получен из множества целевых сигналов. Пространства исходных и целевых сигналов могут являться существенно гетерогенными и обладать разными свойствами.

Определение 6. Общая задача декодирования состоит в построении прогностической модели $\mathbf{f}_{\mathbf{xy}} : \mathbb{R}^{h_x \times m} \times \mathbb{R}^{h_y \times r} \rightarrow \mathbb{R}^{p \times r}$, которая по представлениям предыстории \mathbf{X}_{t,h_x} и \mathbf{Y}_{t,h_y} временных рядов исходных и целевых сигналов предсказывает представление горизонта прогнозирования $\mathbf{Y}_{t,r}$ временных рядов целевых сигналов.

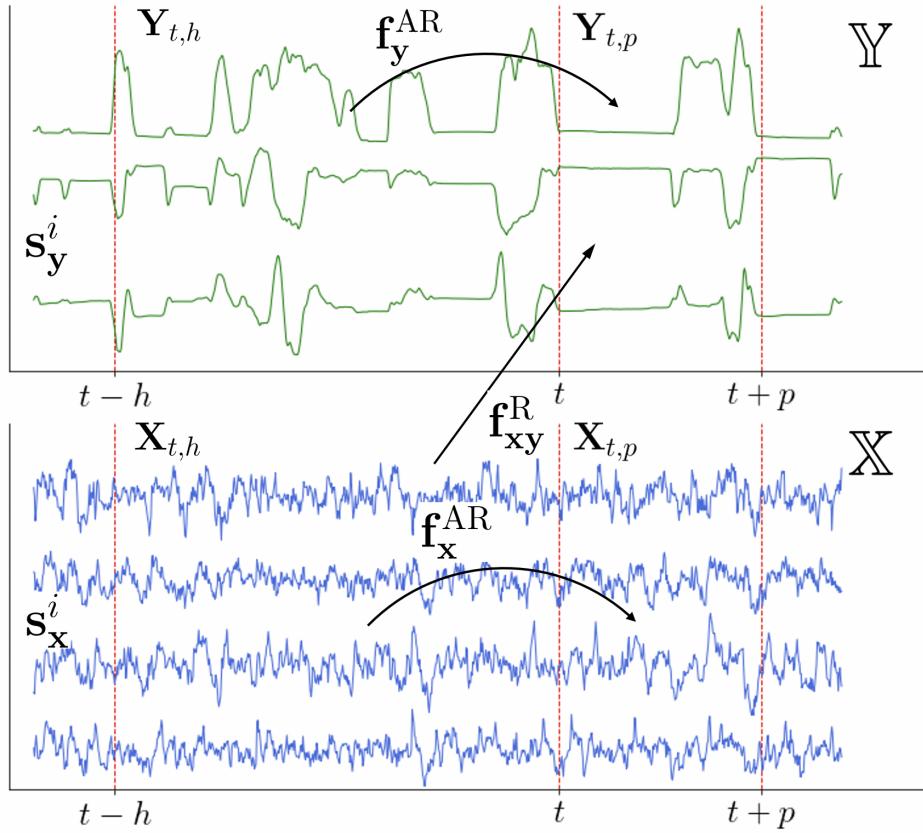


Рис. 1.1: Схема построения моделей декодирования

Отметим, что авторегрессионная модель \mathbf{f}_y^{AR} и регрессионная модель \mathbf{f}_{xy}^R являются частными случаями общей задачи декодирования. А именно, авторегрессионная модель \mathbf{f}_y^{AR} соответствует случаю пустой предыстории временных рядов исходных сигналов (случаю $h_x = 0$), а регрессионная модель \mathbf{f}_{xy}^R соответствует случаю пустой предыстории временных рядов целевых сигналов (случаю $h_y = 0$).

На Рис. 1.1 схематично продемонстрированы принципы построения введенных моделей декодирования временных рядов.

Для построения авторегрессионной модели декодирования временных рядов широко используются два класса линейных методов: авторегрессионные модели и модели скользящего среднего [39, 40]. Авторегрессионные модели AR(p) строят прогноз в виде линейной комбинации p предыдущих значений временного ряда. Модели скользящего среднего MA(q) вместо предыдущих значений временного ряда используют комбинацию ошибок. Модель ARMA(p, q) [41] яв-

ляется комбинацией двух описанных подходов. ARMA(p, q) задает модель как линейную комбинацию p предыдущих значений временного ряда и q предыдущих значений ошибок. Для нахождения оптимальных параметров p и q модели ARMA используются автокорреляционная и частная автокорреляционная функции.

Модель ARMA используется для стационарных временных рядов, отвечающим строгим статистическим предположениям. На практике встречается огромное количество нестационарных временных рядов подверженных тренду, сезонности или цикличности. Модель ARIMA(p, d, q) [41] обобщает модель ARMA для случая нестационарных временных рядов. ARIMA берёт разности порядка d от исходного временного ряда для достижения стационарности данных. При этом на практике оказывается достаточным положить $d = 1$. Заметим, что при $d = 0$ модель ARIMA эквивалентна модели ARMA. Полезным обобщением модели ARIMA является модель AFRIMA [42]. Модель позволяет задать параметр d в виде вещественного числа.

Модель ARIMA плохо справляется с сезонными временными рядами. В работе [39] была предложена модель SARIMA, которая вводит в модель учет сезонной компоненты.

Задача декодирования временных рядов декомпозируется на следующие подзадачи.

- Порождение признакового пространства. Данный этап включает в себя процедуру извлечения признаков из исходных значений сигналов. Процедура порождения признакового пространства может быть основана на экспертных знаниях или же являться моделью машинного обучения. Данная подзадача подробно рассмотрена в главе 6.
- Снижение размерности пространства или выбор признаков. Исходные временные ряды, а также порожденное признаковое пространство оказываются избыточным, что приводит к избыточности и неустойчивости модели. Методы снижения размерности и выбора признаков подробно изложены

в главах 2 и 3.

- Построение модели. После нахождения оптимального низкоразмерного представления исходных данных ставится задача выбора оптимальной модели декодирования.

1.3 Обзор методов снижения размерности для задачи декодирования

Методы снижения размерности позволяют найти низкоразмерное представление исходных данных. Найденное представление используется для построения прогностической модели. При этом метод снижения размерности может учитывать как зависимости в исходной переменной \mathbf{x} , так и в целевой переменной \mathbf{y} .

Метод главных компонент для задачи декодирования. Для устранения линейной зависимости и снижения размерности исходного пространства широко используется метод главных компонент (principal component analysis, PCA). Метод PCA находит низкоразмерное представление матрицы $\mathbf{X} = \mathbf{T}\mathbf{P}$, такое что новое представление $\mathbf{T} \in \mathbb{R}^{m \times l}$ содержит максимальную долю дисперсии исходной матрицы. При этом матрица отображения $\mathbf{P} \in \mathbb{R}^{l \times n}$ ($\mathbf{P}\mathbf{P}^\top = \mathbf{I}$) содержит правые собственные вектора матрицы ковариаций $\mathbf{X}^\top \mathbf{X}$.

Метод PCA является базовым методом снижения размерности пространства. Существует множество модификаций базового метода. Вероятностный PCA [11] рассматривает задачу снижения размерности в терминах вероятностной модели, решая задачу с помощью вариационного EM алгоритма. Разреженный PCA [43] вводит в постановку задачи lasso регуляризацию для того, чтобы сделать матрицу отображения \mathbf{P} разреженной и более интерпретируемой. Нелинейный ядерный PCA [44] отображает исходные данные с помощью нелинейного отображения и использует RKHS для решения исходной задачи.

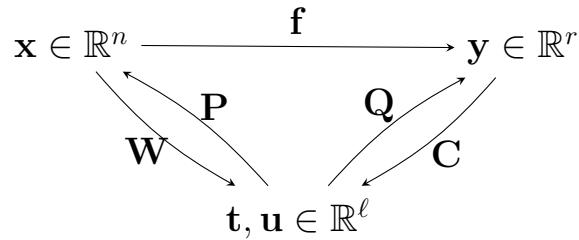
После нахождения матрицы отображения \mathbf{P} задача (1.4) принимает вид

$$\mathcal{L}(\mathbf{B}, \mathbf{T}, \mathbf{Y}) = \left\| \mathbf{Y}_{m \times r} - \mathbf{T}_{m \times l} \cdot \mathbf{B}_{l \times r} \right\|_2^2 \rightarrow \min_{\mathbf{B}}.$$

Модель прогнозирования (1.3) в случае снижения размерности с помощью PCA принимает вид:

$$\mathbf{y} = \mathbf{B}^\top \mathbf{t} + \boldsymbol{\varepsilon} = \mathbf{B}^\top \mathbf{P} \mathbf{x} + \boldsymbol{\varepsilon} = \boldsymbol{\Theta} \mathbf{x} + \boldsymbol{\varepsilon}, \text{ где } \boldsymbol{\Theta} = \mathbf{B}^\top \mathbf{P}.$$

Метод частичных наименьших квадратов для задачи декодирования. Основным недостатком метода PCA является отсутствие учёта взаимосвязи между исходными признаками χ_j и целевыми столбцами ν_j . Метод частичных наименьших квадратов (partial least squares, PLS) проецирует исходную матрицу \mathbf{X} и целевую матрицу в скрытое пространство малой размерностью l ($l < n$). Метод PLS находит в скрытом пространстве матрицы $\mathbf{T}, \mathbf{U} \in \mathbb{R}^{m \times l}$, которые лучше всего описывают исходные матрицы \mathbf{X} и \mathbf{Y} . При этом PLS максимизирует ковариацию между столбцами матриц \mathbf{T} и \mathbf{U} соответственно. Метод PLS соответствует следующей коммутативной диаграмме:



Метод PLS был впервые предложен в работах [7, 45, 46]. Подробное описание алгоритма приведено в работах [47, 48, 49, 50, 51]. В работах [8, 52] приведен обзор обобщений базовой модели PLS. В работе [2] приведена модификация метода PLS для получения разреженного набора признаков.

Исходная матрица \mathbf{X} и целевая матрица \mathbf{Y} проецируются на скрытое про-

пространство следующим образом:

$$\underset{m \times n}{\mathbf{X}} = \underset{m \times l}{\mathbf{T}} \cdot \underset{l \times n}{\mathbf{P}} + \underset{m \times n}{\mathbf{E}_x} = \sum_{k=1}^l \underset{m \times 1}{\boldsymbol{\tau}_k} \cdot \underset{1 \times n}{\mathbf{p}_k^\top} + \underset{m \times n}{\mathbf{E}_x}, \quad (1.5)$$

$$\underset{m \times r}{\mathbf{Y}} = \underset{m \times l}{\mathbf{U}} \cdot \underset{l \times r}{\mathbf{Q}} + \underset{m \times r}{\mathbf{E}_y} = \sum_{k=1}^l \underset{m \times 1}{\boldsymbol{\nu}_k} \cdot \underset{1 \times r}{\mathbf{q}_k^\top} + \underset{m \times r}{\mathbf{E}_y}. \quad (1.6)$$

Здесь \mathbf{T} и \mathbf{U} — образы исходных матриц в скрытом пространстве, причём столбцы матрицы \mathbf{T} ортогональны; \mathbf{P} и \mathbf{Q} — матрицы перехода; \mathbf{E}_x и \mathbf{E}_y — матрицы остатков. Метод PLS восстанавливает линейную зависимость между столбцами матриц \mathbf{T} и \mathbf{U}

$$\mathbf{U} \approx \mathbf{T}\mathbf{B}, \quad \mathbf{B} = \text{diag}(\beta_k), \quad \beta_k = \boldsymbol{\nu}_k^\top \boldsymbol{\tau}_k / (\boldsymbol{\tau}_k^\top \boldsymbol{\tau}_k),$$

где $\{\boldsymbol{\tau}_k\}_{k=1}^l$, $\{\boldsymbol{\nu}_k\}_{k=1}^l$ — столбцы матриц \mathbf{T} и \mathbf{U} соответственно.

Метод решает следующую оптимизационную задачу:

$$\max_{\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1} [\text{cov}(\mathbf{X}\mathbf{p}, \mathbf{Y}\mathbf{q})^2] = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \mathbf{X}^\top \mathbf{Y}\mathbf{q}}{\sqrt{\mathbf{p}^\top \mathbf{p}} \sqrt{\mathbf{q}^\top \mathbf{q}}}. \quad (1.7)$$

Детальное описание алгоритма работы метода PLS с доказательством его корректности приведено в разделе 2.2.

Для демонстрации разницы между методами PCA и PLS был проведен модельный эксперимент для случая, когда размерности пространств исходной и целевой переменных, а также скрытого пространства равны 2 ($n = r = l = 2$). Вектора исходной переменной \mathbf{x}_i сгенерированы из нормального распределения с нулевым матожиданием. Вектора целевой переменной \mathbf{y}_i линейным образом зависят от второй главной компоненты pc_2 матрицы \mathbf{X} и не зависят от первой главной компоненты pc_1 . На Рис. 1.2 показаны результаты работы методов. Синими и зелёными точками изображены вектора исходной переменной \mathbf{x}_i и вектора целевой переменной \mathbf{y}_i . Красным контуром показаны линии уровня матриц ковариаций распределений. Чёрным изображены единичные окружности. Красные стрелки соответствуют главным компонентам матриц \mathbf{X} и \mathbf{Y} . Чёрные стрелки соответствуют векторам матриц \mathbf{W} и \mathbf{C} метода PLS. Данные

матрицы содержат вектора, являющиеся аналогами главных компонент метода PCA. Учёт взаимной связи между матрицами \mathbf{X} и \mathbf{Y} отклоняет вектора \mathbf{w}_k и \mathbf{c}_k от направления главных компонент.

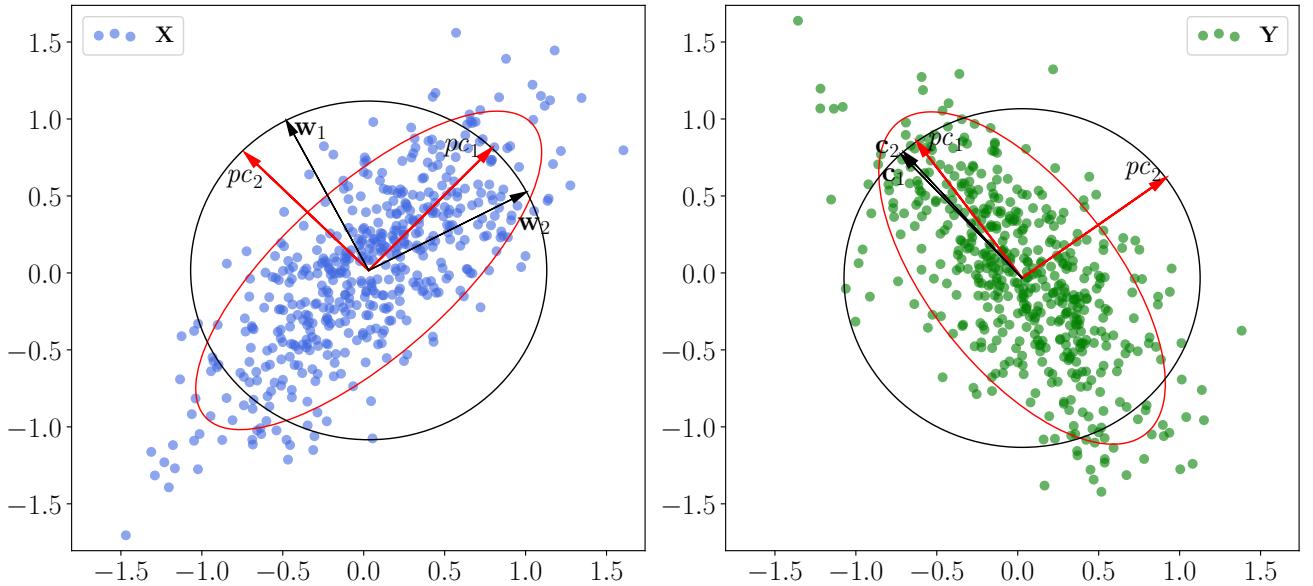


Рис. 1.2: Модельный пример работы методов PCA и PLS

При снижении размерности пространств до одного признака метод PCA выберет первую главную компоненту pc_1 , отбросив компоненту pc_2 , так как первая компонента объясняет большую часть дисперсии исходной матрицы \mathbf{X} . При этом матрица \mathbf{Y} не зависит от pc_1 . Тем самым финальная модель окажется не адекватной. Метод PLS позволяет побороться с данной проблемой.

Канонический анализ корреляций для задачи декодирования. Канонический корреляционный анализ (canonical correlation analysis, CCA) широко применяется для поиска взаимосвязи между двумя наборами переменных [12, 53]. Оптимизационная задача канонического корреляционного анализа (CCA) отличается от задачи PLS (1.7) тем, что вместо максимизации ковариации максимизируется корреляция:

$$\max_{\|\mathbf{p}\|_2 = \|\mathbf{q}\|_2 = 1} [\text{corr}(\mathbf{X}\mathbf{p}, \mathbf{Y}\mathbf{q})^2] = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \mathbf{X}^\top \mathbf{Y} \mathbf{q}}{\sqrt{\mathbf{p}^\top \mathbf{X}^\top \mathbf{X} \mathbf{p}} \sqrt{\mathbf{q}^\top \mathbf{Y}^\top \mathbf{Y} \mathbf{q}}}.$$

Линейная регрессия	PCA	PLS	CCA
0.01	0.24	0.13	0.13

Таблица 1.1: Средняя квадратичная ошибка на модельном примере для методов линейной регрессии, PCA, PLS, CCA

На Рис. 1.3 показан результат работы метода. Основное различие состоит в том, что вектора \mathbf{c}_1 и \mathbf{c}_2 в данном случае становятся ортогональными.

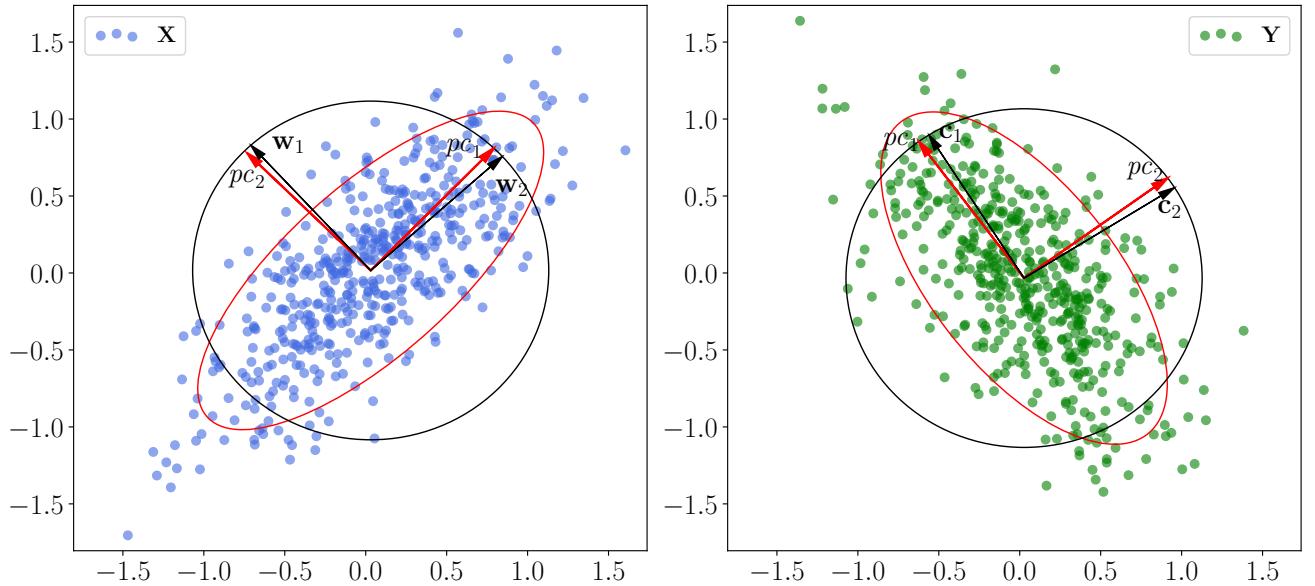


Рис. 1.3: Модельный пример работы методов PCA и CCA

В таблице 1.1 приведены значения квадратичной ошибки $\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$ для методов линейной регрессии, PCA, PLS и CCA. Линейная регрессия отлично справляется с данной задачей. Ошибка метода PCA наибольшая, что подтверждает факт, что для данной ситуации метод не находит нужных зависимостей в пространстве целевой переменной. Методы PLS и CCA показывают схожие результаты.

Нелинейный ядерный CCA [54, 55, 56, 57] является обобщением базового метода. CCA и ядерный CCA широко используются для задач обучения без учителя [58, 59]. Метод имеет область применения от анализа хемометрических [60]

и биологических [61] данных до обработки естественного языка [62, 63], аудио-сигналов [64, 65] и компьютерного зрения [66].

В работе [67] впервые было предложено обобщение метода ССА, работающего с нейросетями. Предлагаемый метод DeepCCA максимизирует корреляцию между представлениями, полученными на выходе нейросети:

$$\max_{\|\mathbf{p}\|_2=\|\mathbf{q}\|_2=1} [\text{corr}(\boldsymbol{\varphi}_x(\mathbf{X}) \cdot \mathbf{p}, \boldsymbol{\varphi}_y(\mathbf{Y}) \cdot \mathbf{q})^2] = \\ = \max_{\mathbf{p}, \mathbf{q}} \frac{\mathbf{p}^\top \boldsymbol{\varphi}_x(\mathbf{X})^\top \boldsymbol{\varphi}_y(\mathbf{Y}) \mathbf{q}}{\sqrt{\mathbf{p}^\top \boldsymbol{\varphi}_x(\mathbf{X})^\top \boldsymbol{\varphi}_x(\mathbf{X}, \mathbf{W}_x) \mathbf{p}} \sqrt{\mathbf{q}^\top \boldsymbol{\varphi}_y(\mathbf{Y})^\top \boldsymbol{\varphi}_y(\mathbf{Y}) \mathbf{q}}}.$$

Здесь $\boldsymbol{\varphi}_x(\mathbf{X})$ и $\boldsymbol{\varphi}_y(\mathbf{Y})$ — нелинейные проекции исходной и целевой матриц. В статье [68] приведен обширный обзор модификаций нейросетевого ССА для работы с многовидовыми данными. С использованием нейросетевых функций модель декодирования способна учитывать существенно нелинейные зависимости как в исходном пространстве, так и в целевом пространстве. Главным недостатком нейросетевого ССА является вычислительная сложность. В работе [69] предложена релаксация исходной функции потерь, которая масштабируется для работы с глубокими моделями нейросетей.

Тензорные линейные методы для задачи декодирования. Если исходная переменная \mathbf{x} является не вектором, а тензором более высокого порядка, то для построения модели тензор может быть вытянут в вектор [70]. В таком случае модель не учитывает имеющиеся зависимости между различными направлениями исходного тензора. Для учета таких зависимостей используются тензорные версии метода PLS [71, 26, 72].

Многомодальные данные в задаче декодирования. Исходная и целевая переменные могут иметь несколько модальностей. Примерами таких модальностей могут быть выровненные аудио и видео [73, 74], аудио и артикуляция [75], изображение и текстовая аннотация [57, 76, 77], параллельный корпус текстов [59, 62, 78, 79].

Если для исходной и целевой переменных имеется более двух модальностей, то для построения скрытого пространства применяются два класса подходов. Первый подход состоит в построении скрытого пространства для каждой пары модальностей [80, 81]. Второй же подход состоит в построении общего единого скрытого пространства для всех модальностей [82, 83].

Глава 2

Задача построения согласованных моделей декодирования

В данной главе приводится формальная постановка задачи построения согласованных моделей декодирования. Вводятся понятия скрытого пространства и процедуры согласования. Доказываются теоремы о выборе оптимальной модели декодирования.

2.1 Процесс согласования моделей в пространстве высокой размерности

Введём предположения о структурах пространств \mathbb{X} и \mathbb{Y} .

Предположение 1. Пусть пространства \mathbb{X} и \mathbb{Y} имеют избыточную размерность. Это означает, что исходная переменная \mathbf{x} и целевая переменная \mathbf{y} принадлежат некоторым многообразиям низкой размерности. В простейшем случае такими многообразиями могут являться вложениями или линейными подпространствами.

Определение 7. Назовём пространство $\mathbb{T} \subset \mathbb{R}^l$ скрытым пространством для пространства $\mathbb{X} \subset \mathbb{R}^n$ ($l \leq n$), если существуют функция $\varphi_{\mathbf{x}} : \mathbb{X} \rightarrow \mathbb{T}$ и функция $\psi_{\mathbf{x}} : \mathbb{T} \rightarrow \mathbb{X}$, такие что

$$\text{для любого } \mathbf{x} \in \mathbb{X} \text{ найдется } \mathbf{t} \in \mathbb{T} : \psi_{\mathbf{x}}(\varphi_{\mathbf{x}}(\mathbf{x})) = \psi_{\mathbf{x}}(\mathbf{t}) = \mathbf{x}.$$

Функцию $\varphi_{\mathbf{x}}(\mathbf{x})$ назовём функцией кодирования переменной \mathbf{x} , функцию $\psi_{\mathbf{x}}(\mathbf{t})$ назовём функцией декодирования.

Аналогично введём определение скрытого пространства $\mathbb{U} \subset \mathbb{R}^s$ для целевого пространства \mathbb{Y} , функции кодирования $\varphi_{\mathbf{y}} : \mathbb{Y} \rightarrow \mathbb{U}$ и декодирования $\psi_{\mathbf{y}} : \mathbb{U} \rightarrow \mathbb{Y}$, такие что

$$\text{для любого } \mathbf{y} \in \mathbb{Y} \text{ найдется } \mathbf{u} \in \mathbb{U} : \psi_{\mathbf{y}}(\varphi_{\mathbf{y}}(\mathbf{y})) = \psi_{\mathbf{y}}(\mathbf{u}) = \mathbf{y}.$$

Образы исходной матрицы \mathbf{X} и целевой матрицы \mathbf{Y} в скрытых пространствах \mathbb{T} и \mathbb{U} имеют вид

$$\mathbf{T} = \varphi_{\mathbf{x}}(\mathbf{X}) = [\mathbf{t}_1, \dots, \mathbf{t}_n]^{\top} = [\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_l],$$

$$\mathbf{U} = \varphi_{\mathbf{y}}(\mathbf{Y}) = [\mathbf{u}_1, \dots, \mathbf{u}_n]^{\top} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_s].$$

Здесь строки $\{\mathbf{t}_i\}_{i=1}^n$ матрицы \mathbf{T} и строки $\{\mathbf{u}_i\}_{i=1}^n$ матрицы \mathbf{U} являются образами векторов исходной переменной $\{\mathbf{x}_i\}_{i=1}^n$ и векторов целевой переменной $\{\mathbf{y}_i\}_{i=1}^n$. Столбцы $\{\boldsymbol{\tau}_j\}_{j=1}^l$ матрицы \mathbf{T} и столбцы $\{\boldsymbol{\nu}_j\}_{j=1}^s$ матрицы \mathbf{U} являются скрытыми векторами.

Определение 8. Скрытые пространства \mathbb{T} и \mathbb{U} являются *согласованными*, если существует *функция связи* $\mathbf{h} : \mathbb{T} \rightarrow \mathbb{U}$, такая что

$$\text{для любого } \mathbf{u} \in \mathbb{U} \text{ найдется } \mathbf{t} \in \mathbb{T} : \mathbf{u} = \mathbf{h}(\mathbf{t}).$$

Предположение 2. Предположим, что в задаче прогнозирования (1.1) пространства \mathbb{T} и \mathbb{U} являются скрытыми для пространств \mathbb{X} и \mathbb{Y} соответственно. Предположим также, что для данных скрытых пространств \mathbb{T} и \mathbb{U} существует функция связи $\mathbf{h} : \mathbb{T} \rightarrow \mathbb{U}$. Тогда выполнено

$$\text{для любого } \mathbf{y} \in \mathbb{Y} \text{ найдется } \mathbf{x} \in \mathbb{X} : \mathbf{y} = \psi_{\mathbf{y}}(\mathbf{u}) = \psi_{\mathbf{y}}(\mathbf{h}(\mathbf{t})) = \psi_{\mathbf{y}}(\mathbf{h}(\varphi_{\mathbf{x}}(\mathbf{x}))),$$

и общая схема задачи поиска согласованной модели декодирования принимает вид коммутативной диаграммы:

$$\begin{array}{ccc} \mathbb{X} \subset \mathbb{R}^n & \xrightarrow{\mathbf{f}} & \mathbb{Y} \subset \mathbb{R}^r \\ \varphi_{\mathbf{x}} \swarrow \psi_{\mathbf{x}} & & \psi_{\mathbf{y}} \searrow \varphi_{\mathbf{y}} \\ \mathbb{T} \subset \mathbb{R}^\ell & \xrightarrow{\mathbf{h}} & \mathbb{U} \subset \mathbb{R}^s \end{array} \tag{2.1}$$

Определение 9. Согласно диаграмме (2.1), определим *согласованную* модель декодирования $\mathbf{f} : \mathbb{X} \rightarrow \mathbb{Y}$ как суперпозицию

$$\mathbf{f} = \psi_{\mathbf{y}} \circ \mathbf{h} \circ \varphi_{\mathbf{x}}. \tag{2.2}$$

Таким образом задача прогнозирования (1.1) сводится к поиску согласованной модели декодирования (2.2). Для поиска оптимальных параметров функций кодирования φ_x и φ_y , декодирования ψ_x и ψ_y , а также функции связи \mathbf{h} ставится задача максимизации *функции согласования*

$$g : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}, \quad g(\boldsymbol{\tau}, \boldsymbol{\nu}) \rightarrow \max_{\varphi_x, \varphi_y, \mathbf{h}}.$$

Каждая пара векторов $\boldsymbol{\tau}, \boldsymbol{\nu}$ ищется последовательно.

Сформулируем примеры методов снижения размерности пространства, описанные в разделе 1.3, в терминах задачи построения согласованной модели декодирования.

Метод главных компонент снижает размерность исходных данных и сохраняет максимальную дисперсию между полученными проекциями. Линейная модель РСА представляет собой ортогональное линейное преобразование исходного признакового пространства в скрытое пространство меньшей размерности.

Функции кодирования $\varphi_x : \mathbb{R}^n \rightarrow \mathbb{R}^l$ и декодирования $\psi_x : \mathbb{R}^l \rightarrow \mathbb{R}^n$ имеют вид

$$\varphi_x(\mathbf{X}) = \underset{m \times n}{\mathbf{X}} \cdot \underset{n \times l}{\mathbf{P}}^\top, \quad \psi_x(\mathbf{T}) = \underset{m \times l}{\mathbf{T}} \cdot \underset{l \times n}{\mathbf{P}},$$

где $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_l]^\top$, при этом $\mathbf{P}\mathbf{P}^\top = \mathbf{I}$.

Скрытые вектора $\boldsymbol{\tau}$ строятся так, чтобы выборочная дисперсия столбцов проекций матрицы \mathbf{X} была максимальной:

$$\mathbf{p} = \arg \max_{\|\mathbf{p}\|_2=1} g(\boldsymbol{\tau}) = \arg \max_{\|\mathbf{p}\|_2=1} [\text{var}(\boldsymbol{\tau})] = \arg \max_{\|\mathbf{p}\|_2=1} [\text{var}(\mathbf{X}\mathbf{p})],$$

где $\text{var}(\boldsymbol{\tau})$ — выборочная дисперсия.

Метод РСА не согласует исходные переменные и целевые переменные. А именно метод РСА не находит функции кодирования φ_y и декодирования ψ_y , а также функцию связи \mathbf{h} . При этом функция согласования $g(\boldsymbol{\tau})$ зависит только от одного аргумента. Из-за этого зависимости в обоих пространствах не учитываются. Пример некорректной работы метода в случае наличия зависимостей как в исходном, так и в целевом пространстве, показан в разделе 1.3.

В методах PLS и CCA функции кодирования и декодирования имеют вид

$$\begin{aligned}\varphi_x(\mathbf{X}) &= \mathbf{XW}, & \varphi_y(\mathbf{Y}) &= \mathbf{YC}, \\ \psi_x(\mathbf{T}) &= \mathbf{TP}^\top, & \psi_y(\mathbf{U}) &= \mathbf{UQ}^\top.\end{aligned}$$

Функция связи \mathbf{h} имеет вид линейной модели, связывающей образы проекций в скрытом пространстве $\mathbf{u} = \mathbf{h}(\mathbf{t}) = \mathbf{B}^\top \mathbf{t}$. В данном случае схема декодирования (2.1) принимает вид следующей коммутативной диаграммы.

$$\begin{array}{ccc} \mathbb{X} \subset \mathbb{R}^n & \xrightarrow{\mathbf{f}} & \mathbb{Y} \subset \mathbb{R}^r \\ \mathbf{W} \swarrow \mathbf{P} & & \mathbf{Q} \searrow \mathbf{C} \\ \mathbb{T} \subset \mathbb{R}^\ell & \xrightarrow{\mathbf{B}} & \mathbb{U} \subset \mathbb{R}^s \end{array}$$

В разделе 2.2 приводится подробная процедура нахождения оптимальных матриц \mathbf{P} , \mathbf{Q} , \mathbf{W} , \mathbf{C} , \mathbf{B} с доказательством корректности.

Различие между методами PLS и CCA заключается в виде функции согласования g . Для метода PLS функция согласования имеет вид $g(\boldsymbol{\tau}, \boldsymbol{\nu}) = \text{cov}(\boldsymbol{\tau}, \boldsymbol{\nu})$, а для метода CCA: $g(\boldsymbol{\tau}, \boldsymbol{\nu}) = \text{corr}(\boldsymbol{\tau}, \boldsymbol{\nu})$.

Помимо линейных моделей декодирования рассматриваются нелинейные методы. В данном случае функции кодирования и декодирования являются нелинейными функциями вида

$$\begin{aligned}\mathbf{t} &= \varphi_x(\mathbf{x}) = \mathbf{W}_x^L \sigma(\dots \sigma(\mathbf{W}_x^2 \sigma(\mathbf{W}_x^1 \mathbf{x})) \dots), \\ \mathbf{u} &= \varphi_y(\mathbf{y}) = \mathbf{W}_y^L \sigma(\dots \sigma(\mathbf{W}_y^2 \sigma(\mathbf{W}_y^1 \mathbf{y})) \dots), \\ \mathbf{x} &= \psi_x(\mathbf{t}) = \mathbf{W}_t^L \sigma(\dots \sigma(\mathbf{W}_t^2 \sigma(\mathbf{W}_t^1 \mathbf{t})) \dots), \\ \mathbf{y} &= \psi_y(\mathbf{u}) = \mathbf{W}_u^L \sigma(\dots \sigma(\mathbf{W}_u^2 \sigma(\mathbf{W}_u^1 \mathbf{u})) \dots).\end{aligned}$$

Каждая функция является нейросетью, т.е. суперпозицией линейных отображений и поэлементных функций активаций.

Требуется найти такие параметры, при которых функция согласования g

достигает своего максимума:

$$g(\boldsymbol{\tau}, \boldsymbol{\nu}) \rightarrow \max_{\mathbf{W}}, \quad (2.3)$$

где $\mathbf{W} = \{\mathbf{W}_x^i, \mathbf{W}_y^i, \mathbf{W}_t^i, \mathbf{W}_u^i\}_{i=1}^L$.

Процесс согласования заключается в максимизации функции согласования $g(\boldsymbol{\tau}, \boldsymbol{\nu})$ по параметрам нейросетей. В работе [67] рассматривается частный случай задачи (2.3). При использовании в качестве функции согласования корреляции между проекциями $g(\boldsymbol{\tau}, \boldsymbol{\nu}) = \text{corr}(\boldsymbol{\tau}, \boldsymbol{\nu})$ частная производная функции согласования по первому аргументу принимает вид

$$\frac{\partial g(\mathbf{T}, \mathbf{U})}{\partial \mathbf{T}} = \frac{1}{m-1} \left(\boldsymbol{\Sigma}_1^{-1/2} \mathbf{U} \mathbf{V}^\top \boldsymbol{\Sigma}_2^{-1/2} \mathbf{U} - \boldsymbol{\Sigma}_1^{-1/2} \mathbf{U} \mathbf{D} \mathbf{V}^\top \boldsymbol{\Sigma}_1^{-1/2} \right),$$

где $\mathbf{U}, \mathbf{D}, \mathbf{V} = \text{SVD}(\boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1^{-1/2} \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_2^{-1/2}$, $\boldsymbol{\Sigma}_1 = \frac{1}{m-1} \mathbf{T} \mathbf{T}^\top$, $\boldsymbol{\Sigma}_2 = \frac{1}{m-1} \mathbf{U} \mathbf{U}^\top$, $\boldsymbol{\Sigma}_{12} = \frac{1}{m-1} \mathbf{T} \mathbf{U}^\top$. Аналогичное выражение имеет частная производная по второму аргументу. Полученное выражение для градиента позволяет построить эффективный алгоритм для решения задачи с использованием градиентных методов оптимизации.

2.2 Доказательство корректности алгоритма проекции в скрытое пространство

Псевдокод алгоритма метода регрессии PLS приведен в алгоритме 1. Алгоритм итеративно на каждом из l шагов вычисляет по одному столбцу $\boldsymbol{\tau}_k, \boldsymbol{\nu}_k$ матриц \mathbf{T}, \mathbf{U} и по одной строке $\mathbf{p}_k, \mathbf{q}_k$ матриц \mathbf{P}, \mathbf{Q} соответственно. После вычисления следующего набора векторов из матриц \mathbf{X}, \mathbf{Y} вычитаются очередные одноранговые аппроксимации. При этом предполагается, что исходные матрицы \mathbf{X} и \mathbf{Y} нормированы (имеют нулевое среднее и единичное среднее отклонение).

Вектора $\boldsymbol{\tau}_k$ и $\boldsymbol{\nu}_k$ из внутреннего цикла алгоритма 1 содержат информацию о исходной матрице \mathbf{X} и целевой матрице \mathbf{Y} соответственно. Блоки из шагов

Algorithm 1 Алгоритм PLS

Вход: $\mathbf{X}, \mathbf{Y}, l$;

Выход: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;

- 1: нормировать матрицы \mathbf{X} и \mathbf{Y} по столбцам
 - 2: инициализировать $\boldsymbol{\nu}_0$ (первый столбец матрицы \mathbf{Y})
 - 3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
 - 4: **для** $k = 1, \dots, l$
 - 5: **повторять**
 - 6: $\mathbf{w}_k := \mathbf{X}_k^\top \boldsymbol{\nu}_{k-1} / (\boldsymbol{\nu}_{k-1}^\top \boldsymbol{\nu}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
 - 7: $\boldsymbol{\tau}_k := \mathbf{X}_k \mathbf{w}_k$
 - 8: $\mathbf{c}_k := \mathbf{Y}_k^\top \boldsymbol{\tau}_k / (\boldsymbol{\tau}_k^\top \boldsymbol{\tau}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
 - 9: $\boldsymbol{\nu}_k := \mathbf{Y}_k \mathbf{c}_k$
 - 10: **пока** $\boldsymbol{\tau}_k$ не стабилизируется
 - 11: $\mathbf{p}_k := \mathbf{X}_k^\top \boldsymbol{\tau}_k / (\boldsymbol{\tau}_k^\top \boldsymbol{\tau}_k), \mathbf{q}_k := \mathbf{Y}_k^\top \boldsymbol{\tau}_k / (\boldsymbol{\tau}_k^\top \boldsymbol{\tau}_k)$
 - 12: $\mathbf{X}_{k+1} := \mathbf{X}_k - \boldsymbol{\tau}_k \mathbf{p}_k^\top$
 - 13: $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \boldsymbol{\tau}_k \mathbf{q}_k^\top$
-

(6)–(7) и шагов (8)–(9) — аналоги метода РСА для матриц \mathbf{X} и \mathbf{Y} [84]. Последовательное выполнение блоков позволяет учесть взаимную связь между матрицами \mathbf{X} и \mathbf{Y} .

Теоретическое обоснование метода PLS следует из следующих утверждений.

Утверждение 1. Максимизация ковариации между векторами $\boldsymbol{\tau}_k$ и $\boldsymbol{\nu}_k$ сохраняет дисперсию столбцов матриц \mathbf{X} и \mathbf{Y} и учитывает их линейную зависимость.

Доказательство. Утверждение следует из равенства

$$\text{cov}(\boldsymbol{\tau}_k, \boldsymbol{\nu}_k) = \text{corr}(\boldsymbol{\tau}_k, \boldsymbol{\nu}_k) \cdot \sqrt{\text{var}(\boldsymbol{\tau}_k)} \cdot \sqrt{\text{var}(\boldsymbol{\nu}_k)}.$$

Максимизация дисперсий векторов $\boldsymbol{\tau}_k$ и $\boldsymbol{\nu}_k$ отвечает за сохранение информации об исходных матрицах, корреляция между векторами отвечает взаимосвязи между \mathbf{X} и \mathbf{Y} . □

Во внутреннем цикле алгоритма 1 вычисляются нормированные вектора весов \mathbf{w}_k и \mathbf{c}_k . Из данных векторов строятся матрицы весов \mathbf{W} и \mathbf{C} соответственно.

Утверждение 2. Вычисленные вектора \mathbf{w}_k и \mathbf{c}_k с помощью итеративной процедуры обновления:

$$\boldsymbol{\tau}_k := \frac{\mathbf{X}_k \mathbf{w}_k}{\|\mathbf{w}_k\|}, \quad \mathbf{w}_k := \frac{\mathbf{X}_k^\top \boldsymbol{\nu}_{k-1}}{\boldsymbol{\nu}_{k-1}^\top \boldsymbol{\nu}_{k-1}}; \quad (2.4)$$

$$\boldsymbol{\nu}_k := \frac{\mathbf{Y}_k \mathbf{c}_k}{\|\mathbf{c}_k\|}, \quad \mathbf{c}_k := \frac{\mathbf{Y}_k^\top \boldsymbol{\tau}_k}{\boldsymbol{\tau}_k^\top \boldsymbol{\tau}_k}. \quad (2.5)$$

будут собственными векторами матриц $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$ и $\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k$, соответствующими максимальным собственным значениям.

Доказательство.

$$\mathbf{w}_k \propto \mathbf{X}_k^\top \boldsymbol{\nu}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \boldsymbol{\tau}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_{k-1},$$

$$\mathbf{c}_k \propto \mathbf{Y}_k^\top \boldsymbol{\tau}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \boldsymbol{\nu}_{k-1} \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1},$$

где символ \propto означает равенство с точностью до мультипликативной константы. Утверждение следует из того факта, что правила обновления векторов \mathbf{w}_k , \mathbf{c}_k совпадают с итерацией алгоритма поиска максимального собственного значения степенным методом. Если матрица \mathbf{A} диагонализуема, \mathbf{x} — некоторый вектор, то

$$\lim_{k \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max},$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы \mathbf{A} , \mathbf{v}_{\max} — собственный вектор матрицы \mathbf{A} , соответствующий $\lambda_{\max}(\mathbf{A})$. \square

Утверждение 3. Вычисленные векторы $\boldsymbol{\tau}_k$ и $\boldsymbol{\nu}_k$ с помощью итеративной процедуры обновления (2.4), (2.5) обладают максимальной ковариацией $\text{cov}(\boldsymbol{\tau}, \boldsymbol{\nu})$.

Доказательство. Максимальная ковариация между векторами $\boldsymbol{\tau}_k$ и $\boldsymbol{\nu}_k$ равна максимальному собственному значению матрицы $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$:

$$\begin{aligned} \max_{\boldsymbol{\tau}_k, \boldsymbol{\nu}_k} \text{cov}(\boldsymbol{\tau}_k, \boldsymbol{\nu}_k)^2 &= \max_{\substack{\|\mathbf{w}_k\|=1 \\ \|\mathbf{c}_k\|=1}} \text{cov}(\mathbf{X}_k \mathbf{w}_k, \mathbf{Y}_k \mathbf{c}_k)^2 = \max_{\substack{\|\mathbf{w}_k\|=1 \\ \|\mathbf{c}_k\|=1}} \text{cov}\left(\mathbf{c}_k^\top \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k\right)^2 = \\ &= \max_{\|\mathbf{w}_k\|=1} \text{cov}\left\|\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\top \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k = \\ &= \lambda_{\max}\left(\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k\right), \end{aligned}$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы \mathbf{A} . Применяя утверждение 2, получаем требуемое. \square

После завершения внутреннего цикла на шаге (11) вычисляются вектора \mathbf{p}_k , \mathbf{q}_k проецированием столбцов матриц \mathbf{X}_k и \mathbf{Y}_k на вектор $\boldsymbol{\tau}_k$. Для перехода на следующий шаг необходимо вычесть из матриц \mathbf{X}_k и \mathbf{Y}_k одноранговые аппроксимации $\boldsymbol{\tau}_k \mathbf{p}_k^\top$ и $\boldsymbol{\tau}_k \mathbf{q}_k^\top$

$$\begin{aligned} \mathbf{X}_{k+1} &= \mathbf{X}_k - \boldsymbol{\tau}_k \mathbf{p}_k^\top = \mathbf{X} - \sum_k \boldsymbol{\tau}_k \mathbf{p}_k^\top, \\ \mathbf{Y}_{k+1} &= \mathbf{Y}_k - \boldsymbol{\tau}_k \mathbf{q}_k^\top = \mathbf{Y} - \sum_k \boldsymbol{\tau}_k \mathbf{q}_k^\top. \end{aligned}$$

При этом каждый следующий вектор $\boldsymbol{\tau}_k$ оказывается ортогонален всем векторам $\{\boldsymbol{\tau}_j\}_{j=1}^{k-1}$.

Теорема 1. В случае линейных функций декодирования $\psi_x(\mathbf{T}) = \mathbf{TP}$, $\psi_y(\mathbf{U}) = \mathbf{UQ}$ и функции согласования $g(\boldsymbol{\tau}, \boldsymbol{\nu}) = \text{cov}(\boldsymbol{\tau}, \boldsymbol{\nu})$ параметры

$$\boldsymbol{\Theta} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1} \mathbf{BQ}$$

являются оптимальными для модели (1.3).

Доказательство. Для получения прогнозов модели и нахождения параметров модели домножим справа формулу (1.5) на матрицу \mathbf{W} . Строки матрицы невязок \mathbf{E} ортогональны столбцам матрицы \mathbf{W} , поэтому

$$\mathbf{XW} = \mathbf{TPW}.$$

Линейное преобразование между векторами в исходном и латентном пространствах имеет вид

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \quad (2.6)$$

где $\mathbf{W}^* = \mathbf{W}(\mathbf{P}\mathbf{W})^{-1}$.

Матрица параметров модели 1.3 находится из уравнений (1.6), (2.6)

$$\mathbf{Y} = \mathbf{T}\mathbf{Q} + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{Q} + \mathbf{E} = \mathbf{X}\Theta + \mathbf{E}. \quad (2.7)$$

Отсюда получаем требуемое выражение. \square

Финальная модель (2.7) является линейной, низкоразмерной в скрытом пространстве. Это снижает избыточность данных и повышает стабильность модели.

2.3 Аддитивная суперпозиция моделей декодирования

Пусть $\mathbf{f}_1(\mathbf{x}_1, \Theta_1)$, $\mathbf{f}_2(\mathbf{x}_2, \Theta_2)$ — линейные модели декодирования сигналов. Рассмотрим аддитивную суперпозицию моделей декодирования.

Определение 10. Назовём *аддитивной суперпозицией* моделей декодирования модель (1.3) вида

$$\mathbf{Y} = \mathbf{f}(\mathbf{x}, \Theta) + \varepsilon = \mathbf{f}_1(\mathbf{x}_1, \Theta_1) + \mathbf{f}_2(\mathbf{x}_2, \Theta_2) + \varepsilon = \Theta_1^\top \mathbf{x}_1 + \Theta_2^\top \mathbf{x}_2 + \varepsilon, \quad (2.8)$$

где вектор $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^n$ состоит из двух подвекторов $\mathbf{x}_1 \in \mathbb{R}^k$, $\mathbf{x}_2 \in \mathbb{R}^{n-k}$. Тем самым матрица параметров $\Theta \in \mathbb{R}^{n \times r}$ состоит из двух подматриц $\Theta_1 \in \mathbb{R}^{k \times r}$, $\Theta_2 \in \mathbb{R}^{(n-k) \times r}$.

Утверждение 4. Оптимальная матрица параметров Θ для модели (2.8), доставляющая минимум функции ошибки (1.4), имеет вид:

$$\Theta_1 = (\mathbf{X}_1^\top \mathbf{M}_{\mathbf{X}_2} \mathbf{X}_1)^{-1} \mathbf{X}_1^\top \mathbf{M}_{\mathbf{X}_2} \mathbf{Y},$$

$$\Theta_2 = (\mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^{-1} \mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{Y},$$

где

$$\begin{aligned}\mathbf{M}_{\mathbf{X}_1} &= (\mathbf{I} - \mathbf{P}_{\mathbf{X}_1}), \quad \mathbf{P}_{\mathbf{X}_1} = \mathbf{X}_1(\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top, \\ \mathbf{M}_{\mathbf{X}_2} &= (\mathbf{I} - \mathbf{P}_{\mathbf{X}_2}), \quad \mathbf{P}_{\mathbf{X}_2} = \mathbf{X}_2(\mathbf{X}_2^\top \mathbf{X}_2)^{-1} \mathbf{X}_2^\top.\end{aligned}$$

Доказательство. Домножим уравнение $\mathbf{Y} = \mathbf{X}\Theta$ слева на матрицу \mathbf{X}^\top

$$\mathbf{X}^\top \mathbf{X}\Theta = \mathbf{X}^\top \mathbf{Y} \Rightarrow \begin{cases} \mathbf{X}_1^\top \mathbf{X}_1 \Theta_1 + \mathbf{X}_1^\top \mathbf{X}_2 \Theta_2 = \mathbf{X}_1^\top \mathbf{Y}, \\ \mathbf{X}_2^\top \mathbf{X}_1 \Theta_1 + \mathbf{X}_2^\top \mathbf{X}_2 \Theta_2 = \mathbf{X}_2^\top \mathbf{Y}. \end{cases}$$

Выразим из этой системы параметры каждой отдельной модели в суперпозиции:

$$\begin{cases} \Theta_1 = (\mathbf{X}_1^\top \mathbf{X}_1)^{-1} \mathbf{X}_1^\top (\mathbf{Y} - \mathbf{X}_2 \Theta_2), \\ \Theta_2 = (\mathbf{X}_2^\top \mathbf{X}_2)^{-1} \mathbf{X}_2^\top (\mathbf{Y} - \mathbf{X}_1 \Theta_1). \end{cases}$$

Подставим полученные выражения для Θ_1 и Θ_2 в исходную систему:

$$\begin{cases} \mathbf{X}_1^\top \mathbf{P}_{\mathbf{X}_1} (\mathbf{Y} - \mathbf{X}_2 \Theta_2) + \mathbf{X}_1^\top \mathbf{X}_2 \Theta_2 = \mathbf{X}_1^\top \mathbf{Y}, \\ \mathbf{X}_2^\top \mathbf{X}_1 \Theta_1 + \mathbf{X}_2^\top \mathbf{P}_{\mathbf{X}_2} (\mathbf{Y} - \mathbf{X}_1 \Theta_1) = \mathbf{X}_2^\top \mathbf{Y}. \end{cases}$$

Выразив матрицы параметров, получим требуемые выражения. \square

Заметим, что матрицы $\mathbf{P}_{\mathbf{X}_1}$ и $\mathbf{P}_{\mathbf{X}_2}$ являются матрицами проекций на подпространства, образованные линейными оболочками столбцов матриц \mathbf{X}_1 и \mathbf{X}_2 соответственно. Таким образом матрицы $\mathbf{M}_{\mathbf{X}_1}$ и $\mathbf{M}_{\mathbf{X}_2}$ являются матрицами проекций на ортогональные подпространства.

Утверждение 5. Оптимальная подматрица Θ_2 в модели (2.8) является решением задачи регрессии

$$\|\mathbf{Y}_1 - \mathbf{X}_{21} \Theta_2\|^2 \rightarrow \min_{\Theta_2}, \tag{2.9}$$

где $\mathbf{Y}_1 = \mathbf{M}_{\mathbf{X}_1} \mathbf{Y}$, $\mathbf{X}_{21} = \mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2$.

Доказательство. Матрица $\mathbf{M}_{\mathbf{X}_1}$ является матрицей проекции на ортогональное подпространство, построенное на линейной оболочке столбцов матрицы \mathbf{X}_1 .

Таким образом, матрица $\mathbf{M}_{\mathbf{X}_1}$ является идемпотентной ($\mathbf{M}_{\mathbf{X}_1}^\top \mathbf{M}_{\mathbf{X}_1} = \mathbf{M}_{\mathbf{X}_1}$). Используя утверждение 4, получаем

$$\begin{aligned}\Theta_2 &= (\mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^{-1} \mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{Y} = (\mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1}^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^{-1} \mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1}^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{Y} = \\ &= ((\mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^\top (\mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^\top)^{-1} (\mathbf{M}_{\mathbf{X}_1} \mathbf{X}_2)^\top (\mathbf{M}_{\mathbf{X}_1} \mathbf{Y}) = (\mathbf{X}_{21}^\top \mathbf{X}_{21})^{-1} \mathbf{X}_{21}^\top \mathbf{Y}_1.\end{aligned}$$

Согласно теореме Гаусса-Маркова Θ_2 является решением задачи регрессии (2.9). \square

Таким образом для нахождении оптимальной модели декодирования $\mathbf{f}_2(\mathbf{x}_2, \Theta_2)$ необходимо спроектировать матрицы \mathbf{Y} и \mathbf{X}_2 на подпространства, ортогональные подпространству, образованному линейной оболочкой столбцов матрицы \mathbf{X}_1 . Похожие результаты были доказаны в эконометрике в работах [85, 86, 87]. Аналогичное утверждение верно и для матрицы Θ_1 .

Утверждение 6. Если в задаче (2.8) $\text{span}(\mathbf{X}_1) \cap \text{span}(\mathbf{X}_2) = \emptyset$, то есть столбцы матрицы \mathbf{X}_1 ортогональны столбцам матрицы \mathbf{X}_2 , то Θ_2 является решением задачи регрессии

$$\|\mathbf{Y} - \mathbf{X}_2 \Theta_2\|^2 \rightarrow \min_{\Theta_2}.$$

Доказательство. Используя факт, что $\mathbf{I} = \mathbf{P}_{\mathbf{X}_1} + \mathbf{M}_{\mathbf{X}_1}$ и $\mathbf{X}_2^\top \mathbf{P}_{\mathbf{X}_1} = \mathbf{0}$, получаем

$$\begin{aligned}(\mathbf{X}_2^\top \mathbf{X}_2)^{-1} \mathbf{X}_2^\top \mathbf{Y} &= (\mathbf{X}_2^\top \mathbf{X}_2)^{-1} \mathbf{X}_2^\top (\mathbf{P}_{\mathbf{X}_1} + \mathbf{M}_{\mathbf{X}_1}) \mathbf{Y} = \\ &= (\mathbf{X}_2^\top \mathbf{X}_2)^{-1} \mathbf{X}_2^\top \mathbf{M}_{\mathbf{X}_1} \mathbf{Y} = (\mathbf{X}_{21}^\top \mathbf{X}_{21})^{-1} \mathbf{X}_{21}^\top \mathbf{Y}_1.\end{aligned}$$

Используя утверждение 5, получаем требуемое утверждение. \square

Данное утверждение показывает, что в случае независимых столбцов матриц \mathbf{X}_1 и \mathbf{X}_2 задача регрессии для аддитивной суперпозиции моделей (2.8) распадается на две независимые подзадачи.

Утверждение 7. Ошибка аддитивной суперпозиции моделей не превышает ошибки каждой из отдельных моделей

$$\mathcal{L}(\Theta^*, \mathbf{X}, \mathbf{Y}) \leq \mathcal{L}(\Theta_1, \mathbf{X}_1, \mathbf{Y}),$$

$$\mathcal{L}(\Theta^*, \mathbf{X}, \mathbf{Y}) \leq \mathcal{L}(\Theta_2, \mathbf{X}_2, \mathbf{Y}).$$

Доказательство. Каждая отдельная модель декодирования является частным случаем аддитивной суперпозиции с параметрами. При параметрах $\Theta = [\Theta_1^\top; \mathbf{0}_{r \times n-k}]^\top$ аддитивная суперпозиция эквивалентна модели $\mathbf{f}_1(\mathbf{x}_1, \Theta_1)$. При параметрах $\Theta = [\mathbf{0}_{r \times k}; \Theta_2^\top]^\top$ аддитивная суперпозиция эквивалентна модели $\mathbf{f}_2(\mathbf{x}_2, \Theta_2)$. Согласно утверждению 4, оптимальные параметры Θ^* доставляют минимум функции ошибки. \square

Утверждение 8. Пусть для аддитивной суперпозиции моделей (2.8) выполнены следующие условия

$$\mathbf{Y} \neq \mathbf{P}_{\mathbf{X}_2} \mathbf{Y}, \quad \mathbf{X}_1 \neq \mathbf{P}_{\mathbf{X}_2} \mathbf{X}_1, \quad \mathbf{Y}^\top \mathbf{M}_{\mathbf{X}_2} \mathbf{X}_1 \neq \mathbf{0}.$$

Тогда выполнено строгое неравенство

$$\mathcal{L}(\Theta^*, \mathbf{X}, \mathbf{Y}) < \mathcal{L}(\Theta_2, \mathbf{X}_2, \mathbf{Y}).$$

Доказательство. Выразим ошибку отдельной модели

$$\mathcal{L}(\Theta_2, \mathbf{X}_2, \mathbf{Y}) = \|\mathbf{Y} - \mathbf{X}_2 \Theta_2\|^2 = \|(\mathbf{I} - \mathbf{P}_{\mathbf{X}_2}) \mathbf{Y}\|^2 = \|\mathbf{M}_{\mathbf{X}_2} \mathbf{Y}\|^2 = \|\hat{\mathbf{Y}}\|^2.$$

Заметим, что

$$\|\mathbf{M}_{\mathbf{X}_2} \mathbf{Y} - \mathbf{M}_{\mathbf{X}_2} \mathbf{X}_1 \Theta_1\|^2 = \|\hat{\mathbf{Y}} - \hat{\mathbf{X}} \Theta_1\|^2 = \|(\mathbf{I} - \mathbf{P}_{\hat{\mathbf{X}}}) \hat{\mathbf{Y}}\|^2 = \|\hat{\mathbf{Y}}\|^2 - \|\mathbf{P}_{\hat{\mathbf{X}}} \hat{\mathbf{Y}}\|^2 \leq \|\hat{\mathbf{Y}}\|^2.$$

При этом неравенство становится строгим при введенных предположениях, так как $\|\mathbf{P}_{\hat{\mathbf{X}}} \hat{\mathbf{Y}}\|^2 > 0$. Используя утверждение 5, получаем требуемое. \square

Рассмотрим случай линейной авторегрессионной модели \mathbf{f}_y^{AR} из определения 4 и линейной регрессионной модели \mathbf{f}_{xy}^R из определения 5. Пусть модель декодирования $\mathbf{f}_{xy} : \mathbb{R}^{h_x \times m} \times \mathbb{R}^{h_y \times r} \rightarrow \mathbb{R}^{p \times r}$ из определения 6 является аддитивной суперпозицией авторегрессионной и регрессионной моделей. Тогда ошибка суперпозиции не будет превышать ошибок авторегрессионной и регрессионной моделей. При этом при условиях, описанных в утверждении 8, ошибка суперпозиции строго меньше каждой отдельной модели. Данное утверждение позволяет

осуществлять выбор моделей в суперпозиции, основанный на анализе проекций подпространств, построенных на линейных оболочках исходных признаковых описаний.

2.4 Анализ линейных методов проекции в скрытое пространство

Для проведения вычислительного эксперимента рассматриваются данные потребления электроэнергии. Временные ряды электроэнергии состоят из почасовых записей (52512 наблюдений). Стока матрицы \mathbf{X} — локальная история сигнала за одну неделю $n = 24 \times 7$. Стока матрицы \mathbf{Y} — локальный прогноз потребления электроэнергии в следующие 24 часа $r = 24$. В этом случае матрицы \mathbf{X} и \mathbf{Y} являются авторегрессионными матрицами.

Вычислительный эксперимент также проводился на данных электрокортикограмм (ECoG) из проекта NeuroTycho [88]. Данные ECoG состоят из 32-канальных сигналов напряжения, снятых с головного мозга. Цель состоит в предсказании по входному сигналу ECoG 3D позиции руки в следующие моменты времени. Исходные сигналы напряжения преобразуются в пространственно-временное представление с помощью вейвлет-преобразования с материнским вейвлетом Морле. Процедура извлечения признаков из исходных данных подробно описана в [89, 72]. Описание исходного сигнала в каждый момент времени имеет размерность 32 (каналы) \times 27 (частоты) = 864. Каждый сигнал представляет собой локальный отрезок времени длительностью $\Delta t = 1s$. Временной шаг между сигналами $\delta t = 0.05s$. Матрицы имеют размеры $\mathbf{X} \in \mathbb{R}^{18900 \times 864}$ и $\mathbf{Y} \in \mathbb{R}^{18900 \times 3k}$, где k — число отсчётов времени прогнозирования. Данные разбиты на тренировочную и тестовую части в соотношении 0,67. Пример исходных сигналов мозга и соответствующей траектории руки показан на Рис. 2.1.

Введём среднеквадратичную ошибку для некоторых матриц $\mathbf{A} = [a_{ij}]$ и $\mathbf{B} =$

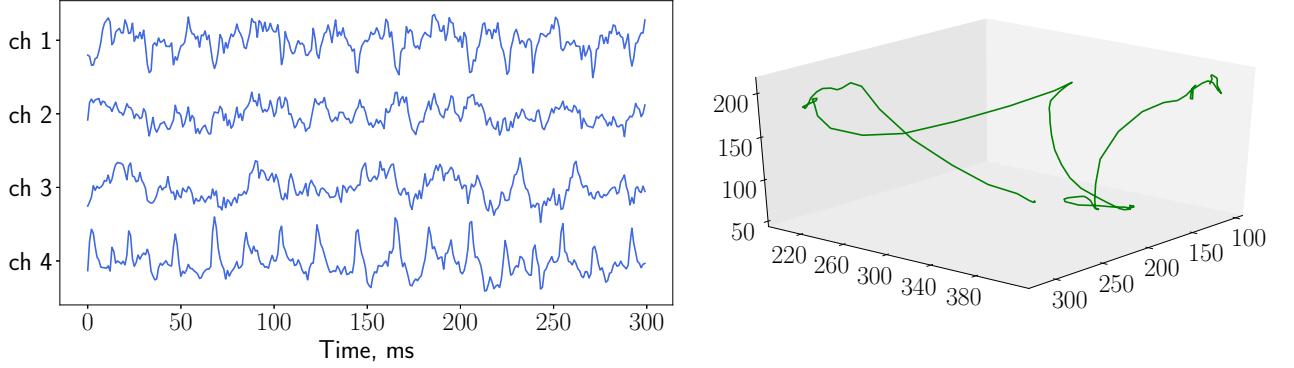


Рис. 2.1: Сигналы мозга (левый график) и 3D координаты руки (правый график)

$$[b_{ij}]$$

$$\text{MSE}(\mathbf{A}, \mathbf{B}) = \sum_{i,j} (a_{ij} - b_{ij})^2.$$

Для оценивания качества аппроксимации вычисляется значение нормированной среднеквадратичной ошибки

$$\text{NMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}})}{\text{MSE}(\mathbf{Y}, \bar{\mathbf{Y}})}, \quad (2.10)$$

где $\hat{\mathbf{Y}}$ — прогноз модели, $\bar{\mathbf{Y}}$ — константный прогноз средним значением по столбцам матрицы.

Результаты на данных электроэнергии. Для нахождения оптимальной размерности l латентного пространства все данные потребления электроэнергии были разбиты на обучающую и валидационную части. Обучающая выборка состоит из 700 временных рядов, валидационная из 370. Зависимость нормированной квадратичной ошибки (2.10) от размерности l латентного пространства представлена на Рис. 2.2. Сначала ошибка резко падает при увеличении размерности скрытого пространства, а затем стабилизируется.

Минимальная ошибка наблюдается при $l = 14$. Построим прогноз потребления электроэнергии при данном l . Результат аппроксимации изображен на Рис. 2.3. Алгоритм PLS восстановил авторегрессионную зависимость и обнару-

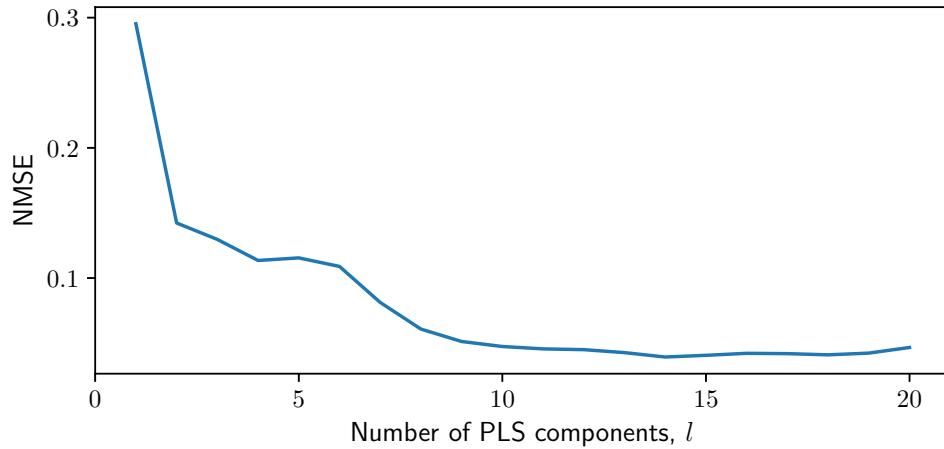


Рис. 2.2: Прогноз потребления электроэнергии методом PLS при размерности латентного пространства $l=14$

жил дневную сезонность.

Результаты на данных электрокортикограммы. На Рис. 2.4 представлена зависимость нормированной квадратичной ошибки (2.10) от размерности латентного пространства. Ошибка аппроксимации меняется незначительно при $l > 5$. Таким образом совместное описание пространственно-временного спектрального представления сигналов и пространственного положения руки может быть представлено вектором размерности $l \ll n$. Зафиксируем $l = 5$. Пример аппроксимации положения руки изображен на Рис. 2.5. Сплошными линиями изображены истинные координаты руки по всем осям, пунктирными линиями показана аппроксимация методом PLS.

2.5 Анализ нелинейных методов проекции в скрытое пространство

Цель вычислительного эксперимента — сравнительный анализ рассматриваемых моделей. Рассматриваются данные, для которых сложность класса линейных методов неадекватно низка. Нелинейные модели позволяют получить точный прогноз при адекватной сложности.

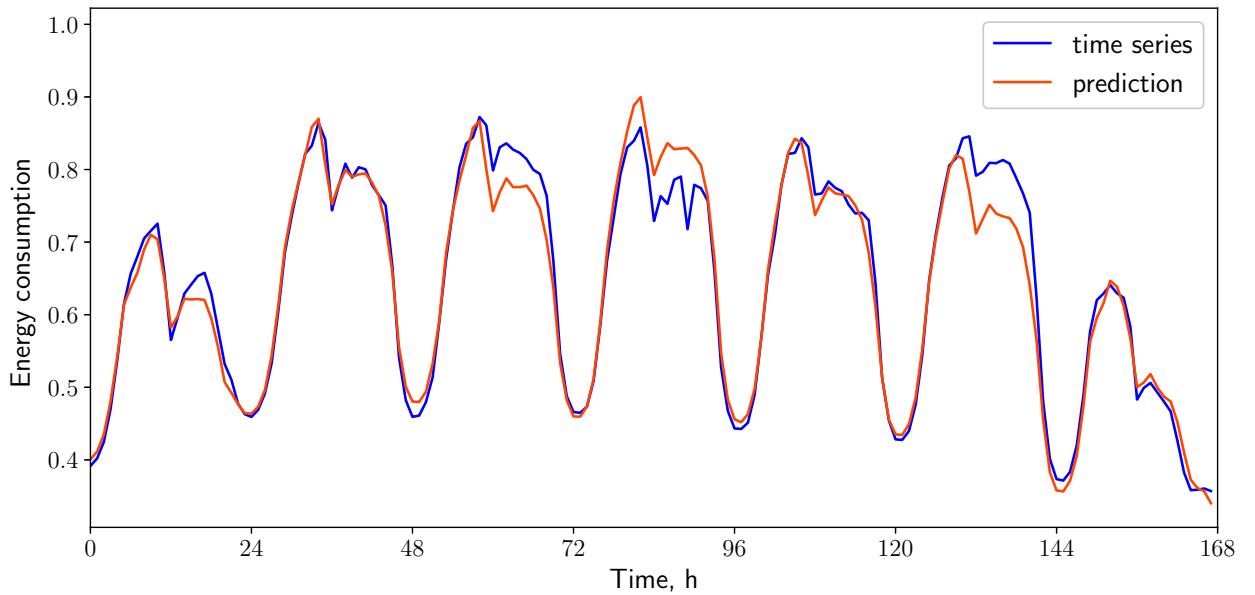


Рис. 2.3: Зависимость ошибки от размерности латентного пространства для данных потребления электроэнергии

Задача фильтрации шума на изображении. Проведем сравнение качества DeepCCA и CCA на задаче классификации зашумленных цифровых изображениях, представленных на Рис. 2.6. Для этого используется набор данных MNIST [90], который состоит из 70 000 цифровых изображений 28×28 образцов рукописного написания цифр. Предлагается получить два новых набора данных \mathbf{X} и \mathbf{Y} следующим образом. Первый набор получается поворотом исходных изображений на угол в диапазоне $[\frac{-\pi}{4}, \frac{\pi}{4}]$. Для получения второго набора данных для каждой картинки из первого набора данных ставится в соответствие случайнym образом картинка с той же цифрой, но с добавлением независимого случайногo шума, распределенного равномерно на отрезке $[0,1]$.

Применив к двум новым наборам данных DeepCCA или CCA, получаем новое низкоразмерное признаковое пространство, которое игнорирует шумы в исходных данных. Модель DeepCCA представляет собой нейронную сеть с $L = 3$ скрытыми слоями. Таким образом, получаем функции кодирования φ_x и φ_y для исходных наборов данных. На новых признаках, полученных разными моделями (DeepCCA и CCA), для первого набора данных, то есть на данных по-

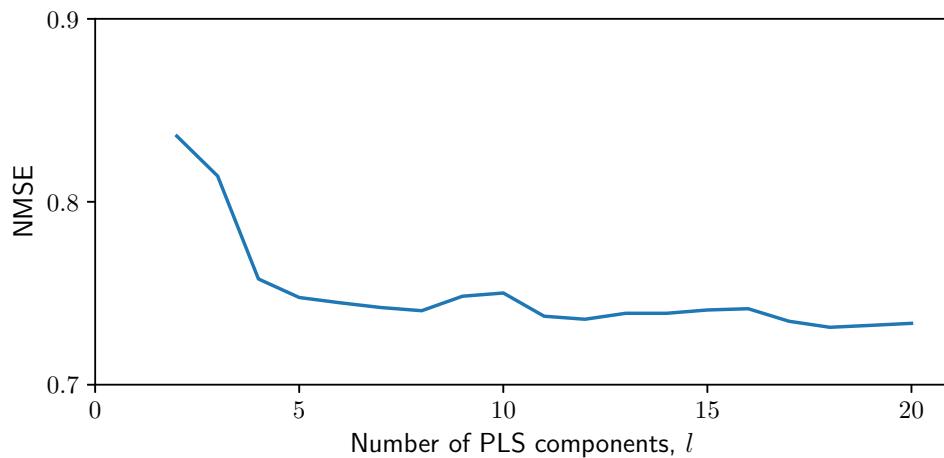


Рис. 2.4: Зависимость ошибки от размерности латентного пространства для данных ECoG

Таблица 2.1: Точность классификации линейного SVM для методов Deep CCA и CCA

Скользящий контроль	Deep CCA ($L = 3$)	CCA
Валидация	92,74%	76,21%
Тест	92,14%	76,07%

сле применения функции кодирования φ_x к первому набору исходных данных, обучим линейный SVM-классификатор. Показателем эффективности будет точность классификации линейного SVM на тестовых данных. В случае построения адекватного скрытого пространства полученные образы изображений будут линейно разделимы. Результаты эксперимента приведены в таблице 2.1. Точность классификации нелинейной модели существенно выше линейного метода CCA.

Задача восстановления изображений. Для анализа процедуры согласования проведен вычислительный эксперимент с предложенными нелинейными моделями. Для снижения размерности пространства используются нейросетевые модели автоэнкодера с согласованием скрытого пространства (2.3). В качестве базовых моделей используются модель автоэнкодера без согласования

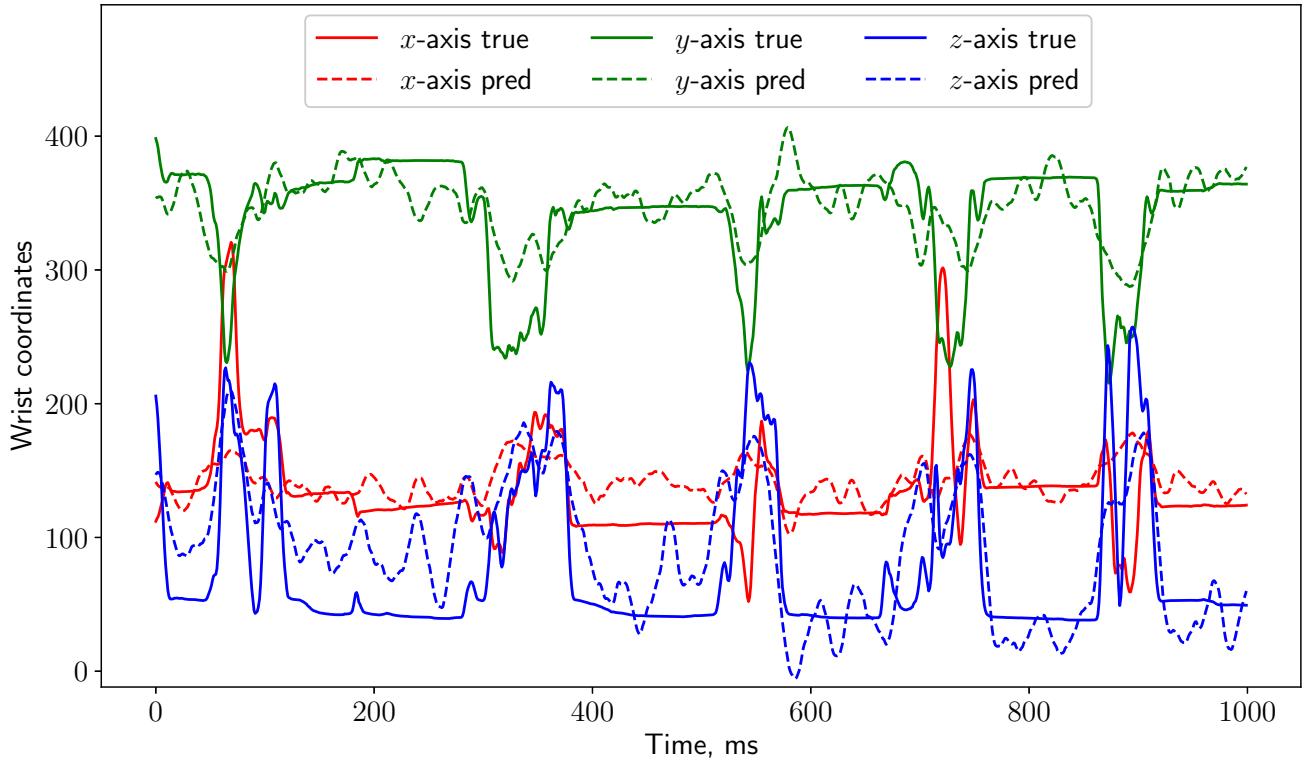


Рис. 2.5: Прогноз движения руки по данным ECoG методом PLS при размерности латентного пространства $l = 5$

скрытых пространств, а также линейный PLS. В качестве исходного набора данных используется набор данных MNIST [90]. Каждое изображение поделено на левую и правую части, как показано на Рис. 2.7. Модель по левому изображению восстанавливает правое изображение.

Модель EncNet1 — нейронная сеть с нелинейными функциями активации, которая обучается на данных после преобразования их автоэнкодером. Модель LinNet1 — нейронная сеть с одним линейным слоем, которая также обучается на преобразованных данных. Для EncNet1 и LinNet1 автоэнкодеры для исходных и целевых изображений используют совместную функцию потерь, которая связывает выходы энкодеров. Модели EncNet2 и LinNet2 устроены аналогично EncNet1 и LinNet1 соответственно, но в автоэнкодерах нет совместной функции потерь. Модель DumbNet — нейронная сеть, которая обучается на исходных данных и имеет такую же структуру, что и EncNet, то есть имеет такое же

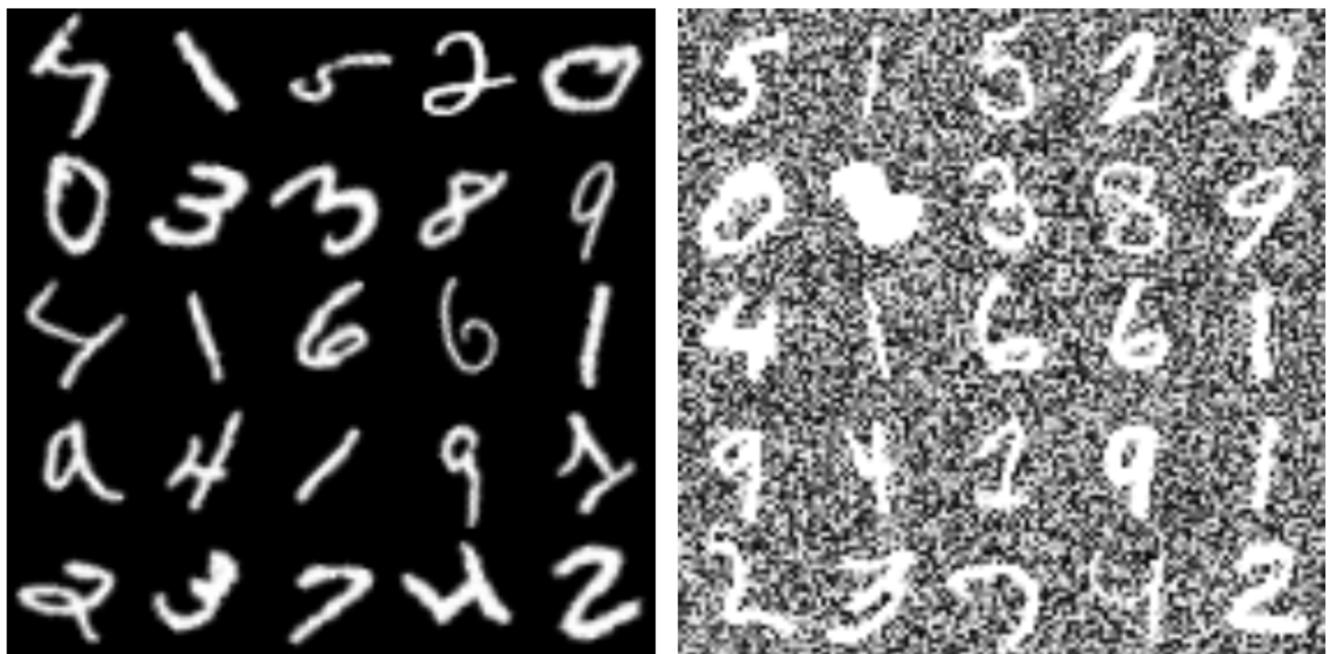


Рис. 2.6: Зашумленные изображения из набора данных MNIST



Рис. 2.7: Набор данных MNIST, в котором каждое изображение разделено пополам

число слоев и в каждом слое такое же количество нейронов, что и у EncNet.

Для оценки качества моделей вычислялась среднеквадратичная ошибка. Примеры восстановленных изображений показаны на Рис. 2.8. Качество моделей, а также их сложность представлены в таблице 2.2. На Рис. 2.8 показано, что предложенные модели EncNet и LinNet позволяют получить более четкие и различимые изображения, в отличие от базовой нелинейной модели DumbNet и линейной модели PLS. Несмотря на заметное улучшение визуального качества изображений, ошибка предложенных моделей выше, чем у модели DumbNet. Это связано с тем, что среднеквадратичная ошибка оказалась неадекватной метрикой в пространстве изображений. Нахождение оптимальной метрики для

оценки качества предложенных методов может быть одним из возможных направлений развития текущего эксперимента.

Таблица 2.2: Квадратичная ошибка для нелинейных моделей в задаче восстановления правой части изображения по левой

	EncNet1	LinNet1	EncNet2	LinNet2	DumbNet	PLS
Число параметров, тыс.	283	239	283	239	283	–
Ошибка на teste	0,147	0,235	0,149	0,236	0,128	0,188

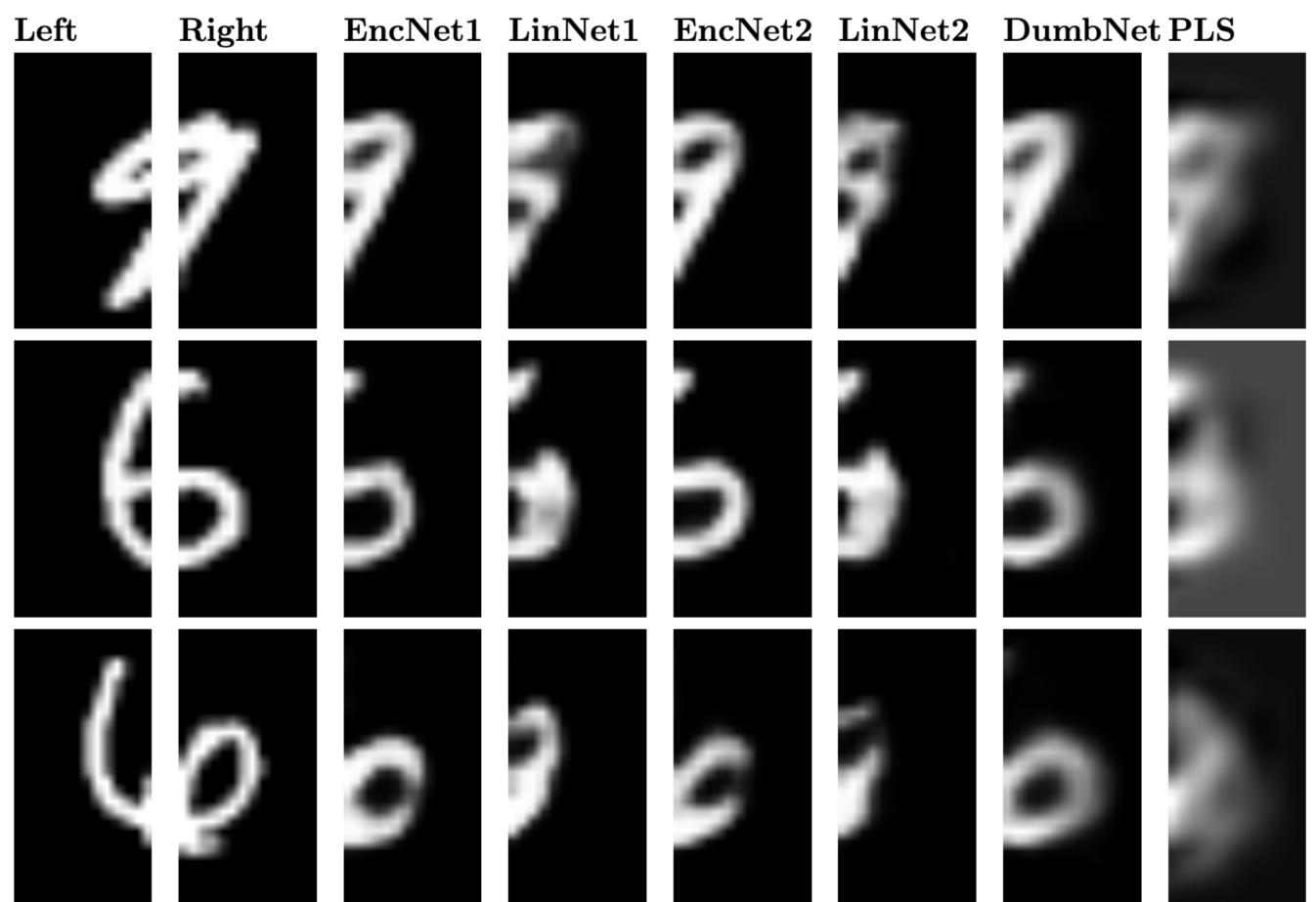


Рис. 2.8: Пример реконструкции правой части изображения по левой для рассматриваемых моделей

Глава 3

Выбор признаков в задаче декодирования сигналов

Задача выбора признаков заключается в поиске оптимального подмножества $\mathcal{A} \subset \{1, \dots, n\}$ индексов признаков среди всех возможных $2^n - 1$ вариантов. Существует взаимооднозначное отображение между подмножеством \mathcal{A} и булевым вектором $\mathbf{a} \in \{0, 1\}^n$, компоненты которого указывают, выбран ли признак. Для нахождения оптимального вектора \mathbf{a} введем функцию ошибки выбора признаков $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Проблема выбора признаков принимает вид:

$$\mathbf{a} = \arg \min_{\mathbf{a}' \in \{0,1\}^n} S(\mathbf{a}', \mathbf{X}, \mathbf{Y}). \quad (3.1)$$

Целью выбора признаков является построение функции $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Конкретные примеры данной функции для рассматриваемых методов выбора признаков приведены ниже и обобщены в таблице 3.1.

Задача (3.1) имеет дискретную область определения $\{0, 1\}^n$. Для решения данной задачи применяется релаксация задачи (3.1) к непрерывной области определения $[0, 1]^n$. Релаксированная задача выбора признаков имеет следующий вид:

$$\mathbf{z} = \arg \min_{\mathbf{z}' \in [0,1]^n} S(\mathbf{z}', \mathbf{X}, \mathbf{Y}). \quad (3.2)$$

Здесь компоненты вектора \mathbf{z} — значения нормированных коэффициентов значимости признаков. Сначала решается задача (3.2), для получения вектора значимостей \mathbf{z} . Затем решение (3.1) восстанавливается с помощью отсечения по порогу следующим образом:

$$\mathbf{a} = [a_j]_{j=1}^n, \quad a_j = \begin{cases} 1, & z_j > \tau; \\ 0, & \text{в противном случае.} \end{cases} \quad (3.3)$$

τ — гиперпараметр, который может быть подобран вручную или выбран с помощью кросс-валидации.

Как только решение \mathbf{a} задачи (3.1) получено, задача (1.4) принимает вид:

$$\mathcal{L}(\Theta_{\mathcal{A}}, \mathbf{X}_{\mathcal{A}}, \mathbf{Y}) = \|\mathbf{Y} - \mathbf{X}_{\mathcal{A}}\Theta_{\mathcal{A}}\|_2^2 \rightarrow \min_{\Theta_{\mathcal{A}}},$$

где индекс \mathcal{A} обозначает подматрицу со столбцами, индексы которых содержатся в \mathcal{A} .

3.1 Выбор признаков с помощью квадратичного программирования

Если между столбцами исходной матрицы \mathbf{X} существует линейная зависимость, то решение задачи линейной регрессии

$$\|\mathbf{v} - \mathbf{X}\theta\|_2^2 \rightarrow \min_{\theta \in \mathbb{R}^n}. \quad (3.4)$$

оказывается неустойчивым. Методы выбора признаков находят подмножество $\mathcal{A} \in \{1, \dots, n\}$ оптимальных столбцов матрицы \mathbf{X} .

Метод QPFS [14] выбирает некоррелированные признаки, релевантные целевому столбцу \mathbf{v} . Чтобы формализовать этот подход, введем две функции: Sim(\mathbf{X}) и Rel(\mathbf{X}, \mathbf{v}). Sim(\mathbf{X}) контролирует избыточность между признаками, Rel(\mathbf{X}, \mathbf{v}) содержит релевантности между каждым признаком и целевым столбцом. Мы хотим минимизировать функцию Sim и максимизировать Rel одновременно.

QPFS предлагает явный способ построения функций Sim и Rel. Метод минимизирует следующую функцию ошибки

$$\underbrace{\mathbf{z}^\top \mathbf{Qz}}_{\text{Sim}} - \alpha \cdot \underbrace{\mathbf{b}^\top \mathbf{z}}_{\text{Rel}} \rightarrow \min_{\substack{\mathbf{z} \in \mathbb{R}_+^n \\ \|\mathbf{z}\|_1=1}}. \quad (3.5)$$

Элементы матрицы парных взаимодействий $\mathbf{Q} \in \mathbb{R}^{n \times n}$ содержат коэффициенты попарного сходства между признаками. Вектор релевантностей признаков $\mathbf{b} \in \mathbb{R}^n$ выражает сходство между каждым признаком и целевым столбцом \mathbf{v} . Нормированный вектор \mathbf{z} отражает значимость каждого признака. Функция ошибки (3.5) штрафует зависимые признаки функцией Sim и штрафует признаки, не релевантные к целевой переменной функцией Rel. Параметр α

позволяет контролировать компромисс между Sim и Rel. Авторы оригинальной статьи QPFS [14] предложили способ выбора α , чтобы уравновесить вклад членов $\text{Sim}(\mathbf{X})$ и $\text{Rel}(\mathbf{X}, \mathbf{v})$

$$\alpha = \frac{\bar{\mathbf{Q}}}{\bar{\mathbf{Q}} + \bar{\mathbf{b}}}, \quad \text{где } \bar{\mathbf{Q}} = \text{mean}(\mathbf{Q}), \quad \bar{\mathbf{b}} = \text{mean}(\mathbf{b}).$$

Чтобы выделить оптимальное подмножество признаков, применяется отсечение по порогу (3.3).

Для измерения сходства используется выборочный коэффициент корреляции Пирсона между парами признаков для функции Sim, и между признаками и целевым столбцом для функции Rel:

$$\mathbf{Q} = [|\text{corr}(\chi_i, \chi_j)|]_{i,j=1}^n, \quad \mathbf{b} = [|\text{corr}(\chi_i, \mathbf{v})|]_{i=1}^n. \quad (3.6)$$

Здесь

$$\text{corr}(\chi, \mathbf{v}) = \frac{\sum_{i=1}^m (\chi_i - \bar{\chi})(\mathbf{v}_i - \bar{\mathbf{v}})}{\sqrt{\sum_{i=1}^m (\chi_i - \bar{\chi})^2 \sum_{i=1}^m (\mathbf{v}_i - \bar{\mathbf{v}})^2}}.$$

Другие способы определения \mathbf{Q} и \mathbf{b} рассматриваются в [5]. В работе [5] показано, что метод QPFS превосходит многие существующие методы выбора признаков на различных внешних критериях качества.

Задача (3.5) является выпуклой, если матрица \mathbf{Q} является неотрицательно определенной. В общем случае это не всегда верно. Чтобы удовлетворить этому условию спектр матрицы \mathbf{Q} смещается, и матрица \mathbf{Q} заменяется на $\mathbf{Q} - \lambda_{\min} \mathbf{I}$, где λ_{\min} является минимальным собственным значением \mathbf{Q} .

3.2 Методы выбора признаков для случая векторной целевой переменной

В данном разделе описаны предлагаемые методы выбора признаков для случая векторной целевой переменной. В этом случае компоненты целевой переменной коррелируют между собой. Предлагаются методы, учитывающие зависимости как в исходном, так и в целевом пространствах.

Агрегация релевантностей целевых столбцов. В работе [1], чтобы применить метод QPFS к векторному случаю ($r > 1$), релевантности признаков агрегируются по всем r компонентам целевой переменной. Член $\text{Sim}(\mathbf{X})$ остаётся без изменений, матрица парных взаимодействий \mathbf{Q} определяется как (3.6). Вектор релевантностей \mathbf{b} агрегируется по всем компонентам целевой переменной и определяется как

$$\mathbf{b} = \left[\sum_{k=1}^r |\text{corr}(\boldsymbol{\chi}_i, \mathbf{v}_k)| \right]_{i=1}^n.$$

Недостатком такого подхода является отсутствие учёта зависимостей в столбцах матрицы \mathbf{Y} . Рассмотрим следующий пример:

$$\mathbf{X} = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3], \quad \mathbf{Y} = [\underbrace{\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_1}_{r-1}, \mathbf{v}_2].$$

Пусть матрица \mathbf{X} содержит 3 столбца, матрица \mathbf{Y} — r столбцов, где первые $r - 1$ компонент целевой переменной идентичны. Попарные сходства признаков задаются матрицей \mathbf{Q} . Матрица \mathbf{B} содержит попарные сходства признаков и целевых столбцов. Вектор \mathbf{b} получен суммированием матрицы \mathbf{B} по столбцами

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.8 \\ 0 & 0.8 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.4 & \dots & 0.4 & 0 \\ 0.5 & \dots & 0.5 & 0.8 \\ 0.8 & \dots & 0.8 & 0.1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (r-1) \cdot 0.4 + 0 \\ (r-1) \cdot 0.5 + 0.8 \\ (r-1) \cdot 0.8 + 0.1 \end{bmatrix}. \quad (3.7)$$

Пусть необходимо выбрать только 2 признака. В данном случае оптимальным подмножеством признаков является $[\boldsymbol{\chi}_1, \boldsymbol{\chi}_2]$. Признак $\boldsymbol{\chi}_2$ предсказывает второй целевой столбец \mathbf{v}_2 , комбинация признаков $\boldsymbol{\chi}_1, \boldsymbol{\chi}_2$ прогнозирует первый целевой столбец \mathbf{v}_1 . Метод QPFS для $r = 2$ дает решение $\mathbf{z} = [0.37, 0.61, 0.02]$. Это совпадает с описанным решением. Однако, если добавить коллинеарные столбцы в матрицу \mathbf{Y} и увеличить r до 5, то решением QPFS будет $\mathbf{z} = [0.40, 0.17, 0.43]$. Здесь потерян признак $\boldsymbol{\chi}_2$ и выбран избыточный признак $\boldsymbol{\chi}_3$. В следующих подразделах предлагаются обобщения метода QPFS, которые позволяют бороться с проблемой данного примера.

Симметричный учёт значимости признаков и целевых переменных. Чтобы учесть зависимости в столбцах матрицы \mathbf{Y} , обобщим функцию QPFS (3.5) для случая векторной целевой переменной ($r > 1$). Добавим член $\text{Sim}(\mathbf{Y})$ и изменим член $\text{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z}_y = 1}} . \quad (3.8)$$

Определим элементы матриц $\mathbf{Q}_x \in \mathbb{R}^{n \times n}$, $\mathbf{Q}_y \in \mathbb{R}^{r \times r}$ и $\mathbf{B} \in \mathbb{R}^{n \times r}$ следующим образом:

$$\mathbf{Q}_x = [|\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)|]_{i,j=1}^n, \quad \mathbf{Q}_y = [|\text{corr}(\boldsymbol{v}_i, \boldsymbol{v}_j)|]_{i,j=1}^r, \quad \mathbf{B} = [|\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{v}_j)|]_{i=1,\dots,n, j=1,\dots,r}.$$

Вектор \mathbf{z}_x содержит коэффициенты значимости признаков, \mathbf{z}_y — коэффициенты значимости целевых столбцов. Коррелированные целевые столбцы штрафуются членом $\text{Sim}(\mathbf{Y})$ и получают более низкие значения значимости.

Коэффициенты α_1 , α_2 , и α_3 контролируют влияние каждого члена на функцию (3.8) и удовлетворяют следующим условиям:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_i \geq 0, i = 1, 2, 3.$$

Утверждение 9. Баланс между $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Sim}(\mathbf{Y})$ в задаче (3.8) достигается при:

$$[\alpha_1, \alpha_2, \alpha_3] = \frac{1}{\overline{\mathbf{Q}}_y \overline{\mathbf{B}} + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}} [\overline{\mathbf{Q}}_y \overline{\mathbf{B}}, \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y, \overline{\mathbf{Q}}_x \overline{\mathbf{B}}]. \quad (3.9)$$

Доказательство. Значения α_1 , α_2 , и α_3 получаются путем решения следующих уравнений:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1,$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}} = \alpha_3 \overline{\mathbf{Q}}_y.$$

Здесь $\overline{\mathbf{Q}}_x$, $\overline{\mathbf{B}}$ и $\overline{\mathbf{Q}}_y$ — средние значения соответствующих матриц \mathbf{Q}_x , \mathbf{B} и \mathbf{Q}_y членов $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Sim}(\mathbf{Y})$. \square

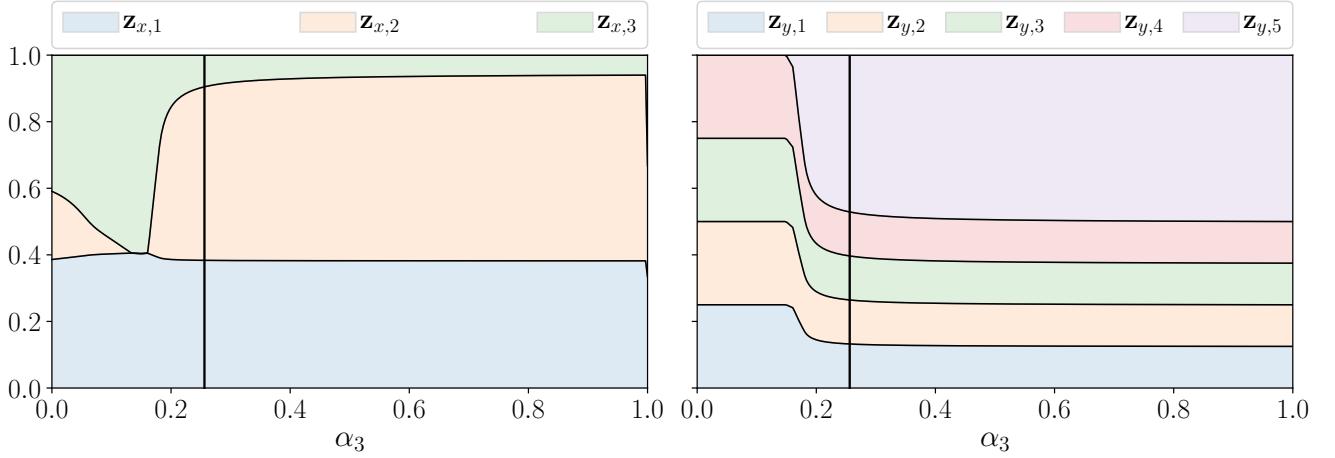


Рис. 3.1: Значимости признаков \mathbf{z}_x и целевых столбцов \mathbf{z}_y в зависимости от α_3 для рассмотренного примера

Для изучения зависимости $\text{Sim}(\mathbf{Y})$ на функцию (3.8), зафиксируем соотношение между α_1 и α_2 :

$$\alpha_1 = \frac{(1 - \alpha_3)\bar{\mathbf{B}}}{\bar{\mathbf{Q}}_x + \bar{\mathbf{B}}}, \quad \alpha_2 = \frac{(1 - \alpha_3)\bar{\mathbf{Q}}_x}{\bar{\mathbf{Q}}_x + \bar{\mathbf{B}}}, \quad \alpha_3 \in [0, 1]. \quad (3.10)$$

Применим предложенный метод к приведенному примеру (3.7). Матрица \mathbf{Q} соответствует матрице \mathbf{Q}_x . Определим матрицы \mathbf{Q}_y как $\text{corr}(\mathbf{v}_1, \mathbf{v}_2) = 0.2$, а все остальные элементы зададим 1. Рисунок 3.1 показывает значение векторов значимостей признаков \mathbf{z}_x и целевых столбцов \mathbf{z}_y в зависимости от значения коэффициента α_3 . Если α_3 мало, значимости всех целевых столбцов не различимы и значимость признака χ_3 выше значимости признака χ_2 . При увеличении α_3 до 0.2, коэффициент значимости $\mathbf{z}_{y,5}$ целевого столбца \mathbf{v}_5 увеличивается наряду со значимостью признака χ_2 .

Минимаксная постановка задачи выбора признаков. Функция (3.8) является симметричной по отношению к \mathbf{z}_x и \mathbf{z}_y . Она штрафует признаки, которые коррелированы и не имеют отношения к целевым столбцам. Кроме того, она штрафует целевые столбцы, которые коррелированы между собой и недостаточно коррелируют с признаками. Это приводит к малым значениям значимостей для целевых столбцов, которые слабо коррелируют с признаками, и большим

значениям для целевых столбцов, которые сильно коррелируют с признаками. Этот результат противоречит интуиции. Цель — предсказать все целевые столбцы, особенно те, которые слабо коррелируют с признаками. Сформулируем две взаимосвязанные задачи:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}}, \quad (3.11)$$

$$\alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} + \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_y \geq \mathbf{0}_r, \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}}. \quad (3.12)$$

Разница между (3.11) и (3.12) заключается в знаке перед членом Rel. В пространстве исходной переменной нерелевантные признаки должны иметь меньшие значения значимости. В то же время целевые столбцы, не релевантные признакам, должны иметь большую значимость. Задачи (3.11) и (3.12) объединяются в совместную минимакс или максмин постановку

$$\min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} f(\mathbf{z}_x, \mathbf{z}_y), \quad \left(\text{или } \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} f(\mathbf{z}_x, \mathbf{z}_y) \right), \quad (3.13)$$

где

$$f(\mathbf{z}_x, \mathbf{z}_y) = \alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} - \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})}.$$

Теорема 2. Для положительно определенной матрицы \mathbf{Q}_x и \mathbf{Q}_y , максмин и минимакс задачи (3.13) имеют одинаковое оптимальное значение.

Доказательство. Введём обозначения

$$\mathbb{C}^n = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z} = 1\}, \quad \mathbb{C}^r = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z} = 1\}.$$

Множества \mathbb{C}^n и \mathbb{C}^r компактные и выпуклые. Функция $f : \mathbb{C}^n \times \mathbb{C}^r \rightarrow \mathbb{R}$ является непрерывной. Если \mathbf{Q}_x и \mathbf{Q}_y положительно определены, функция f является выпукло-вогнутой. Таким образом $f(\cdot, \mathbf{z}_y) : \mathbb{C}^n \rightarrow \mathbb{R}$ выпуклая при фиксированном \mathbf{z}_y , а $f(\mathbf{z}_x, \cdot) : \mathbb{C}^r \rightarrow \mathbb{R}$ вогнута при фиксированном \mathbf{z}_x . В этом

случае по теореме Неймана о минимаксе

$$\min_{\mathbf{z}_x \in \mathbb{C}^n} \max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\mathbf{z}_y \in \mathbb{C}^r} \min_{\mathbf{z}_x \in \mathbb{C}^n} f(\mathbf{z}_x, \mathbf{z}_y).$$

□

Утверждение 10. Минимаксная задача (3.13) эквивалентна задаче квадратичного программирования с $n + r + 1$ переменными.

Доказательство. Для решения минимакс задачи (3.13), зафиксируем некоторый $\mathbf{z}_x \in \mathbb{C}^n$. Для фиксированного вектора \mathbf{z}_x решаем задачу

$$\max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]. \quad (3.14)$$

Лагранжиан для данной задачи:

$$L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y + \lambda \cdot (\mathbf{1}_r^\top \mathbf{z}_y - 1) + \boldsymbol{\mu}^\top \mathbf{z}_y.$$

Здесь вектор множителей Лагранжа $\boldsymbol{\mu}$, который соответствует ограничениям на неравенства $\mathbf{z}_y \geq \mathbf{0}_r$, является неотрицательным. Двойственной задачей является

$$\min_{\lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \min_{\lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} \left[\max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) \right]. \quad (3.15)$$

Для задачи квадратичного программирования (3.14) с положительно определенными матрицами \mathbf{Q}_x и \mathbf{Q}_y выполняются условия сильной двойственности. Таким образом, оптимальное значение (3.14) равно оптимальному значению (3.15). Это позволяет перейти от решения задачи (3.13) к решению задачи

$$\min_{\mathbf{z}_x \in \mathbb{C}^n, \lambda, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_y, \lambda, \boldsymbol{\mu}).$$

Полагая градиент $\nabla_{\mathbf{z}_y} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})$ равным нулю, получим оптимальное значение \mathbf{z}_y :

$$\mathbf{z}_y = \frac{1}{2\alpha_3} \mathbf{Q}_y^{-1} \left(-\alpha_2 \cdot \mathbf{B}^\top \mathbf{z}_x + \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu} \right).$$

Двойственная функция принимает вид

$$\begin{aligned} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) &= \max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \mathbf{z}_x^\top \left(-\frac{\alpha_2^2}{4\alpha_3} \cdot \mathbf{B} \mathbf{Q}_y^{-1} \mathbf{B}^\top - \alpha_1 \cdot \mathbf{Q}_x \right) \mathbf{z}_x \\ &\quad - \frac{1}{4\alpha_3} \lambda^2 \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \mathbf{1}_r - \frac{1}{4\alpha_3} \cdot \boldsymbol{\mu}^\top \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \lambda \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \mathbf{B}^\top \mathbf{z}_x \\ &\quad - \frac{1}{2\alpha_3} \lambda \cdot \mathbf{1}_r^\top \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \cdot \boldsymbol{\mu}^\top \mathbf{Q}_y^{-1} \mathbf{B}^\top \mathbf{z}_x + \lambda. \end{aligned}$$

Данная функция является квадратичной формой с $n + r + 1$ переменными. \square

Несимметричный учёт значимостей признаков и целевых переменных. Естественным способом преодоления проблемы метода SymImp является добавление штрафа для целевых столбцов, которые коррелируют с признаками. Добавим линейный член $\mathbf{b}^\top \mathbf{z}_y$ в член $\text{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\underbrace{\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \underbrace{\alpha_2 \cdot \left(\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \mathbf{b}^\top \mathbf{z}_y \right)}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} + \underbrace{\alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r, \mathbf{1}_r^\top \mathbf{z}_y = 1}}. \quad (3.16)$$

Утверждение 11. Пусть вектор \mathbf{b} равен

$$b_j = \max_{i=1, \dots, n} [\mathbf{B}]_{i,j}.$$

Тогда значение коэффициентов значимостей вектора \mathbf{z}_y будут неотрицательными в $\text{Rel}(\mathbf{X}, \mathbf{Y})$ для задачи (3.16).

Доказательство. Утверждение следует из факта

$$\sum_{i=1}^n z_i b_{ij} \leq \left(\sum_{i=1}^n z_i \right) \max_{i=1, \dots, n} b_{ij} = \max_{i=1, \dots, n} b_{ij},$$

где $z_i \geq 0$ и $\sum_{i=1}^n z_i = 1$. \square

Следовательно, функция (3.16) штрафует в меньшей мере признаки, которые имеют отношение к целевым столбцам, и целевые столбцы, которые недостаточно коррелированы с признаками.

Утверждение 12. Баланс между членами $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$ и $\text{Rel}(\mathbf{X}, \mathbf{Y})$ для задачи (3.16) достигается при следующих коэффициентах:

$$[\alpha_1, \alpha_2, \alpha_3] = \frac{1}{\overline{\mathbf{Q}}_y (\overline{\mathbf{b}} - \overline{\mathbf{B}}) + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}} [\overline{\mathbf{Q}}_y (\overline{\mathbf{b}} - \overline{\mathbf{B}}), \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y, \overline{\mathbf{Q}}_x \overline{\mathbf{B}}].$$

Доказательство. Необходимые значения α_1 , α_2 , и α_3 являются решением следующей системы уравнений:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1,$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}}, \quad (3.17)$$

$$\alpha_2 (\overline{\mathbf{b}} - \overline{\mathbf{B}}) = \alpha_3 \overline{\mathbf{Q}}_y. \quad (3.18)$$

Здесь, в (3.17) уравновешены $\text{Sim}(\mathbf{X})$ с первым слагаемым $\text{Rel}(\mathbf{X}, \mathbf{Y})$, а в (3.18) уравновешены $\text{Sim}(\mathbf{Y})$ с $\text{Rel}(\mathbf{X}, \mathbf{Y})$. \square

Теорема 3. В случае скалярной целевой переменной ($r = 1$) предлагаемые методы выбора признаков SymImp (3.8), MinMax (3.13), AsymImp (3.16) совпадают с оригинальным методом QPFS (3.5).

Доказательство. Если r равно 1, то $\mathbf{Q}_y = q_y$ — скаляр, $\mathbf{z}_y = 1$ и $\mathbf{B} = \mathbf{b}$. Задачи (3.8), (3.13) и (3.16) принимают вид

$$\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{b} \rightarrow \min_{\mathbf{z}_x \geq \mathbf{0}_n, \mathbf{1}_n^\top \mathbf{z}_x = 1}.$$

При $\alpha = \frac{\alpha_2}{\alpha_1 + \alpha_2}$ последняя задача принимает вид (3.5). \square

Таблица 3.1 демонстрирует основные задачи и функции ошибок для каждого метода. RelAgg является базовой стратегией и не учитывает корреляции в целевом пространстве. SymImp штрафует попарные корреляции между целевыми столбцами. MinMax более чувствителен к целевым столбцам, которые трудно предсказать. Стратегия Asymimp добавляет линейный член к функции SymImp, чтобы сделать вклад признаков и целевых столбцов асимметричным.

Таблица 3.1: Обзор предлагаемых обобщений метода QPFS для векторной целевой переменной

Метод	Задача	Функция ошибки $S(\mathbf{z} \mathbf{X}, \mathbf{Y})$
RelAgg	$\min[\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$	$\min_{\mathbf{z}_x} [(1 - \alpha) \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{1}_r]$
SymImp	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x, \mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y + \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$
MinMax	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$ $\max [\text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x} \max_{\mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$
AsymImp	$\min [\text{Sim}(\mathbf{X}) - \text{Rel}(\mathbf{X}, \mathbf{Y})]$ $\max [\text{Rel}(\mathbf{X}, \mathbf{Y}) + \text{Sim}(\mathbf{Y})]$	$\min_{\mathbf{z}_x, \mathbf{z}_y} [\alpha_1 \cdot \mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot (\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y - \mathbf{b}^\top \mathbf{z}_y) + \alpha_3 \cdot \mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y]$

3.3 Анализ методов учета значимостей целевых переменных

Внешние критерии качества. Для оценки предложенных методов выбора признаков, введём критерии оценки качества выбранного подмножества признаков. Определим коэффициент мультикорреляции как среднее значение коэффициента множественной корреляции следующим образом:

$$R^2 = \frac{1}{r} \text{tr} \left(\mathbf{C}^\top \mathbf{R}^{-1} \mathbf{C} \right), \quad \text{где } \mathbf{C} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{v}_j)]_{i=1, \dots, n, j=1, \dots, r}, \quad \mathbf{R} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)]_{i,j=1}^n.$$

Этот коэффициент принимает значение между 0 и 1. Большее значение R^2 соответствует лучшему подмножеству признаков.

Нормированная среднеквадратичная ошибка (sRMSE) отображает качество прогнозирования модели. Оценка sRMSE считается на тренировочной и тестовой выборке.

$$\text{sRMSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}}) = \sqrt{\frac{\text{MSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}})}{\text{MSE}(\mathbf{Y}, \overline{\mathbf{Y}})}} = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}_{\mathbf{a}}\|_2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_2}.$$

Здесь $\widehat{\mathbf{Y}}_{\mathbf{a}} = \mathbf{X}_{\mathbf{a}} \boldsymbol{\Theta}_{\mathbf{a}}^\top$ — прогноз модели, $\overline{\mathbf{Y}}$ — предсказание константной модели,

полученное усреднением целевой переменной по всем компонентам. Данный показатель на тестовой выборке необходимо минимизировать.

Байесовский информационный критерий (BIC) — компромисс между качеством предсказания и размером выбранного подмножества признаков $\|\mathbf{a}\|_0 = \#\{j : a_j \neq 0\} = \sum_{j=1}^n a_j$:

$$\text{BIC} = m \ln \left(\text{MSE}(\mathbf{Y}, \hat{\mathbf{Y}}_{\mathbf{a}}) \right) + \|\mathbf{a}\|_0 \cdot \ln m.$$

Чем меньше значение BIC, тем лучше набор признаков.

Данные. Вычислительный эксперимент проводился на данных электрокортиограмм. Описание данных приведено в разделе 2.4.

На Рис. 3.2 показаны матрицы корреляций для исходных матриц \mathbf{X} и \mathbf{Y} данных ECoG. Частоты в матрице \mathbf{X} сильно коррелированы. В целевой матрице \mathbf{Y} корреляции между осями несущественны по сравнению с корреляциями между последовательными моментами времени и эти корреляции спадают со временем.

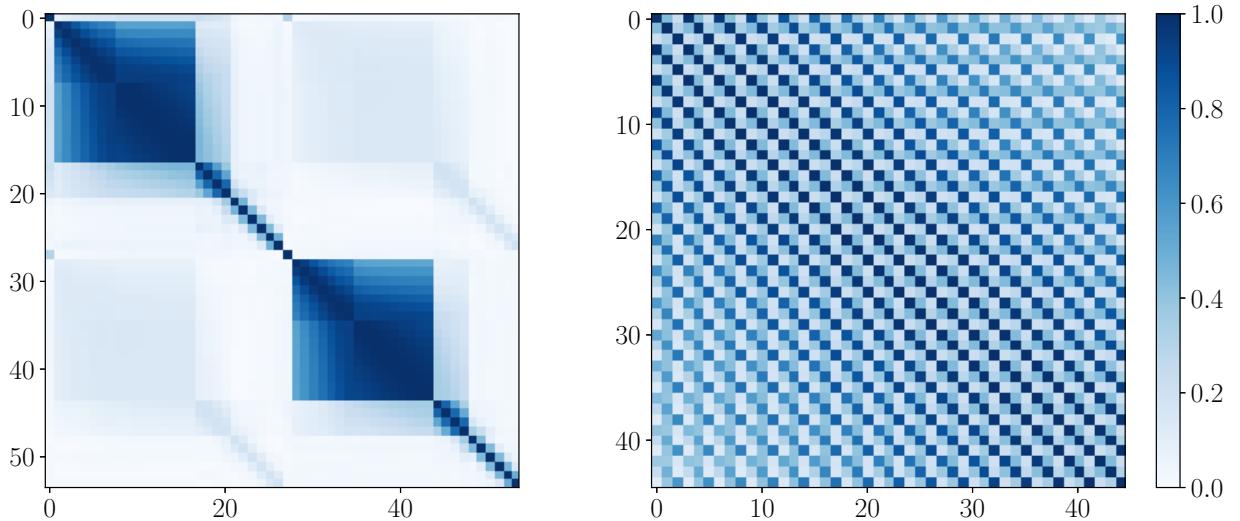


Рис. 3.2: Матрицы корреляций для матрицы плана \mathbf{X} и целевой матрицы \mathbf{Y} для данных ECoG

Результаты. Применим метод SymImp QPFS для различных значений коэффициента α_3 согласно формуле (3.10). Зависимость значимостей целевых столбцов \mathbf{z}_y относительно коэффициента α_3 для различных значений k показана на Рис. 3.3. Значимости целевых столбцов почти одинаковы для всех координат запястья при прогнозировании одного отсчёта времени ($k = 1$), что отражает независимость между координатами x , y и z . Для $k = 2$ и $k = 3$ значимости некоторых целевых столбцов становятся нулевыми при увеличении α_3 . Вертикальные линии соответствуют оптимальному значению α_3 , вычисленному по (3.9). При этом значении α_3 значения компонент \mathbf{z}_y совпадают. Таким образом, метод не учитывает различия между целевыми столбцами для $k = 1, 2, 3$.

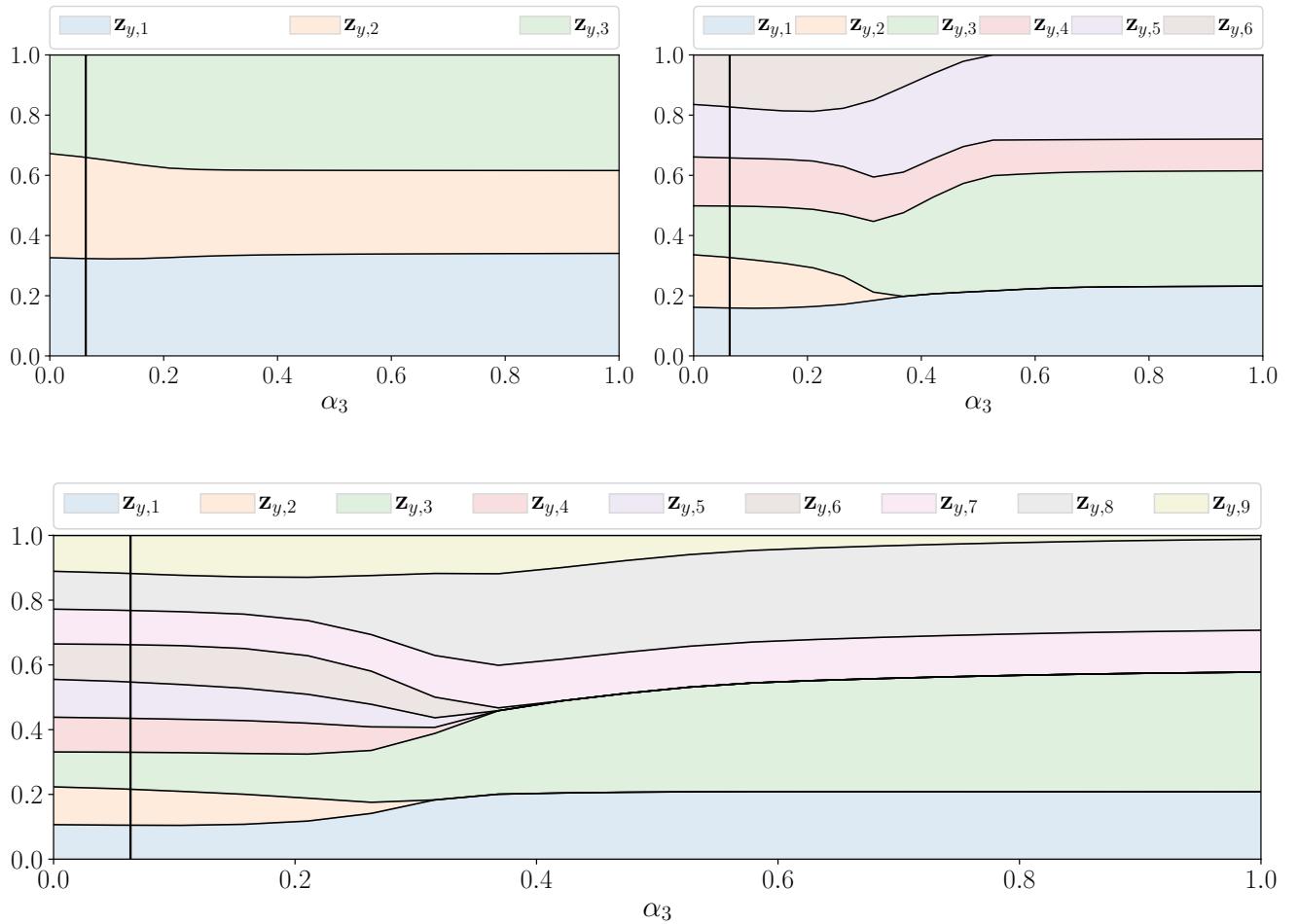


Рис. 3.3: Значимости целевых столбцов \mathbf{z}_y в зависимости от α_3 для метода SymImp QPFS

Предлагаемые методы QPFS для случая векторной целевой переменной,

приведенные в таблице 3.1 применяются для набора данных ECoG. Решим задачу выбора признаков для каждого из методов, чтобы получить вектора значимостей признаков. Отсортируем по убыванию признаки по значению их значимостей. Обучим линейную модель, постепенно добавляя в неё признаки. Исследуются значения описанных критериев качества при увеличении количества отобранных признаков. На Рис. 3.4 показаны результаты для случая прогнозирования $k = 30$ отсчётов времени. Порог значимости признаков τ обозначен цветными тиками. Пороговые значения τ для предлагаемых методов больше, чем для базового метода RelAgg. Метод SymImp имеет большой порог, не позволяя получить малый набор признаков. Однако метод SymImp обладает наилучшей предсказательной способностью с точки зрения sRMSE на тестовых данных. Второй по качеству результат по sRMSE показал метод AsymImp. Все предложенные методы достигают меньшей ошибки на тестовой выборке по сравнению с методом RelAgg. Критерий устойчивости также выше для предложенных методов. Метод AsymImp показывает лучшие результаты с точки зрения качества прогнозирования и размера выбранного подмножества признаков.

Чтобы сравнить структуру выбранных подмножеств признаков и исследовать стабильность процедуры выбора признаков, используется метод генерации данных с помощью бутстрепа. Генерируется множество подвыборок, выбирая объекты по одному с возвращениями. Затем решается задача выбора признаков для каждой пары исходной матрицы \mathbf{X} и целевой матрицы \mathbf{Y} . Сравниваются полученные вектора значимостей для различных подвыборок данных. В качестве меры стабильности работы методов вычисляется средний попарный коэффициент корреляции Спирмена и попарное ℓ_2 расстояние. В таблице 3.2 показана средняя ошибка sRMSE, размер подмножества признаков и описанные статистики для каждого метода. Ошибка считалась на обученной линейной модели с использованием 50 признаков с наибольшими значениями значимостей. Asymimp дает наименьшую ошибку на тестовой выборке. Размер выбранных подмножеств признаков завышен при использовании порогового зна-



Рис. 3.4: Сравнение предложенных методов выбора признаков для данных ECoG при прогнозировании $k = 30$ отсчётов времени

чения $\tau = 10^{-4}$.

Таблица 3.2: Стабильность предложенных методов выбора признаков

	sRMSE	$\ \mathbf{a}\ _0$	Spearman ρ	ℓ_2
RelAgg	0.965 ± 0.002	26.8 ± 3.8	0.915 ± 0.016	0.145 ± 0.018
SymImp	0.961 ± 0.001	224.4 ± 9.0	0.910 ± 0.017	0.025 ± 0.002
MinMax	0.961 ± 0.002	101.0 ± 2.1	0.932 ± 0.009	0.059 ± 0.004
AsymImp	0.955 ± 0.001	85.8 ± 10.2	0.926 ± 0.011	0.078 ± 0.007

Для того, чтобы сравнить методы снижения размерности и выбора признаков, используется модель PLS, описанная в главе 2. На Рис. 3.5 показана ошибка sRMSE на тренировочной и тестовой выборках в зависимости от размерности

скрытого пространства l . Ошибка на тестовой выборке достигает минимума при $l = 11$. Метод PLS приводит к меньшей ошибке, но модель не является разреженной.

На Рис. 3.6 приведено сравнение 3 моделей: линейной регрессии; регрессии PLS, построенной на 100 признаках QPFS; регрессии PLS со всеми признаками. Линейная регрессия со всеми признаками не рассматривается, так как ее результаты близки к константному прогнозу. На рисунке также приведены результаты методов lasso и elastic net, которые широко используются для выбора признаков. В данном эксперименте использовался метод Asymimp QPFS. Размерность скрытого пространства PLS $l = 15$. Результаты регрессии PLS значительно лучше, линейной регрессии с признаками QPFS. Это означает, что последняя модель не является достаточно гибкой. Тем не менее, лучший результат показывает модель PLS, построенная на признаках QPFS. Данная модель является разреженной, так как использует только 100 исходных признаков. Способность модели PLS находить оптимальное скрытое представление данных улучшает предсказательную способность модели.

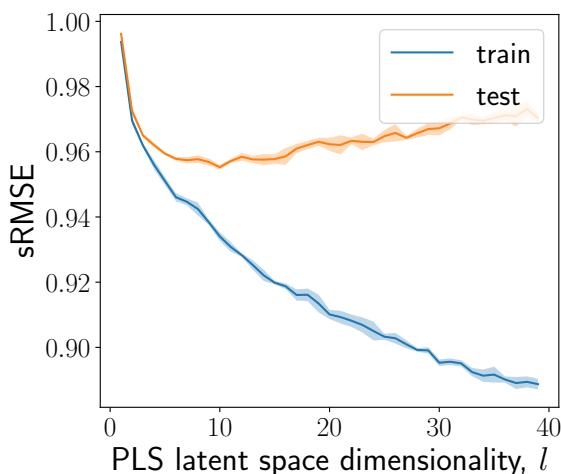


Рис. 3.5: Ошибка sRMSE на тестовой выборке для модели PLS

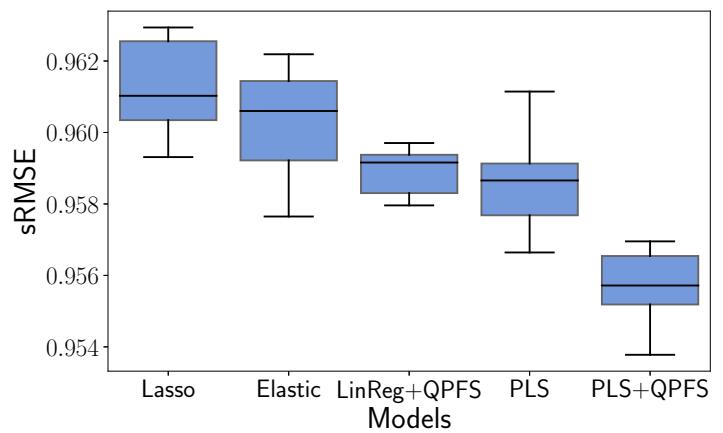


Рис. 3.6: Диаграммы размаха значений sRMSE на тестовой выборке для моделей Lasso, Elastic, LinReg+QPFS, PLS, PLS+QPFS

Глава 4

Выбор параметров нелинейных моделей с помощью квадратичного отбора признаков

Функция ошибки для моделей с большим числом параметров имеет сложный ландшафт с многими локальными минимумами. В этом случае алгоритм оптимизации приводит к разным решениям в зависимости от инициализации исходных параметров.

Алгоритм оптимизации представляет собой итерационный процесс. На каждом шаге для получения следующего приближения параметров модели обновляются текущие параметры. Разработано множество алгоритмов оптимизации первого порядка, использующих вектор первых производных функции ошибки. Наиболее известными алгоритмами являются градиентный спуск, метод момента Нестерова [91], AdaGrad [92], Adam [93]. Данные алгоритмы используются для оптимизации глубоких нейронных сетей [94]. Метод Ньютона — алгоритм второго порядка, использующий матрицу вторых производных функции ошибки. Метод Ньютона находит обновления параметров для квадратичной аппроксимации функции ошибки и сходится за адекватное число итераций. Недостатком методов оптимизации второго порядка является огромная и плохо обусловленная матрица Гессиана. Процесс оптимизации в этом случае расходится и является вычислительно дорогостоящим. Авторы [95, 96] предлагают аппроксимации для матрицы Гессиана и регуляризацию для решения этой проблемы. В статье [97] метод Ньютона применяется к глубоким нейронным сетям.

В данной главе приводится анализ параметров модели, которые не находятся в оптимуме. Приводится метод выбора активных параметров модели, основанный на методе QPFS, который подробно описан в главе 3. Рассматриваются задачи нелинейной регрессии с квадратичной функцией потерь, логистической регрессии с кросс-энтропийной функцией потерь.

4.1 Задача выбора параметров для оптимизации нелинейных моделей

Модель $f(\mathbf{x}, \boldsymbol{\theta})$ с параметрами $\boldsymbol{\theta} \in \mathbb{R}^p$ предсказывает целевую переменную $y \in \mathbb{Y}$ по исходной переменной $\mathbf{x} \in \mathbb{R}^n$. Пространство \mathbb{Y} представляет собой бинарные метки классов $\{0, 1\}$ для задачи двухклассовой классификации и \mathbb{R} для задачи регрессии. Даны исходная матрица $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times n}$ и целевой столбец $\mathbf{y} = [y_1, \dots, y_m]^\top \in \mathbb{Y}^m$. Цель состоит в нахождении оптимальных параметров $\boldsymbol{\theta}^*$. Параметры $\boldsymbol{\theta}$ вычисляются минимизацией функции ошибки:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}). \quad (4.1)$$

Данная задача полностью соответствует рассмотренной задаче (1.2) для случая скалярной целевой переменной ($r = 1$). В качестве функции ошибки $\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$ рассматриваются квадратичная ошибка для задачи регрессии:

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}) = \frac{1}{2} \|\mathbf{y} - \mathbf{f}(\mathbf{X}, \boldsymbol{\theta})\|^2 = \frac{1}{2} \sum_{i=1}^m (y_i - f(\mathbf{x}_i, \boldsymbol{\theta}))^2, \quad (4.2)$$

и функция кросс-энтропии для задачи бинарной классификации:

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}) = \sum_{i=1}^m [y_i \log f(\mathbf{x}_i, \boldsymbol{\theta}) + (1 - y_i) \log(1 - f(\mathbf{x}_i, \boldsymbol{\theta}))]. \quad (4.3)$$

Задача (4.1) решается с помощью итеративной процедуры оптимизации. Для получения параметров на шаге k текущие параметры $\boldsymbol{\theta}^{k-1}$ обновляются по следующему правилу:

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \Delta \boldsymbol{\theta}^{k-1}. \quad (4.4)$$

Для выбора вектора обновлений $\Delta \boldsymbol{\theta}$ используется метод оптимизации Ньютона.

Метод Ньютона нестабилен и вычислительно сложен. В работе предлагается стабильный метод Ньютона. Перед шагом градиента предлагается выбрать подмножество активных параметров модели, которые оказывают наибольшее влияние на функцию ошибки $\mathcal{L}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})$. Введём определение активного параметра модели, используя необходимое условие оптимальности первого порядка.

Определение 11. Параметр θ_j для модели $f(\mathbf{x}, \boldsymbol{\theta})$ является *активным*, если $\mathbf{J}^\top(\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{y}) \neq 0$.

Подробный вывод условия из определения приводится в разделе 4.3. Обновление параметров производится только для отобранного множества индексов $\mathcal{A} = \{j : a_j = 1, \mathbf{a} \in \{0, 1\}^p\}$

$$\begin{aligned}\boldsymbol{\theta}_{\mathcal{A}}^k &= \boldsymbol{\theta}_{\mathcal{A}}^{k-1} + \Delta\boldsymbol{\theta}_{\mathcal{A}}^{k-1}, & \boldsymbol{\theta}_{\mathcal{A}} &= \{\theta_j : j \in \mathcal{A}\}, \\ \boldsymbol{\theta}_{\bar{\mathcal{A}}}^k &= \boldsymbol{\theta}_{\bar{\mathcal{A}}}^{k-1}, & \boldsymbol{\theta}_{\bar{\mathcal{A}}} &= \{\theta_j : j \notin \mathcal{A}\}.\end{aligned}$$

Чтобы выбрать оптимальное подмножество индексов \mathcal{A} , из всех возможных $2^p - 1$ подмножеств, вводится функция ошибки

$$\mathbf{a} = \arg \min_{\mathbf{a}' \in \{0, 1\}^p} S(\mathbf{a}', \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}), \quad (4.5)$$

аналогичная функции ошибки (3.1) для задачи выбора признаков. Задача (4.5) решается на каждом шаге k процесса оптимизации для текущих параметров $\boldsymbol{\theta}^k$.

Метод QPFS используется для решения задачи (4.5). QPFS выбирает подмножество параметров \mathbf{a} для вектора обновлений $\Delta\boldsymbol{\theta}$, которые оказывают наибольшее влияние на вектор остатков и являются попарно независимыми. Функция ошибки (3.5) соответствует функции ошибки $S(\mathbf{a}, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$

$$\mathbf{a} = \arg \max_{\mathbf{a}' \in \{0, 1\}^p} S(\mathbf{a}', \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}) \Leftrightarrow \arg \min_{\mathbf{a} \in \mathbb{R}_+^p, \|\mathbf{a}\|_1=1} [\mathbf{a}^\top \mathbf{Q} \mathbf{a} - \alpha \cdot \mathbf{b}^\top \mathbf{a}].$$

В работе показано, что для модели нелинейной регрессии с квадратичной функцией ошибки (4.2) и для модели логистической регрессии с крос-энтропией (4.3), каждый шаг оптимизации эквивалентен задаче линейной регрессии (3.4).

4.2 Метод Ньютона для оптимизации параметров

Метод Ньютона использует условие оптимизации первого порядка для задачи (4.1) и линеаризует градиент $S(\boldsymbol{\theta})$:

$$\nabla S(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) = \nabla S(\boldsymbol{\theta}) + \mathbf{H} \cdot \Delta\boldsymbol{\theta} = \mathbf{0},$$

$$\Delta\boldsymbol{\theta} = -\mathbf{H}^{-1}\nabla S(\boldsymbol{\theta}).$$

где $\mathbf{H} = \nabla^2 S(\boldsymbol{\theta})$ является матрицей Гессиана функции ошибки $S(\boldsymbol{\theta})$.

Итерация (4.4) метода Ньютона имеет вид

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} - \mathbf{H}^{-1}\nabla S(\boldsymbol{\theta}). \quad (4.6)$$

На каждой итерации требуется обращать матрицу Гессиана \mathbf{H} . Мерой плохой обусловленности для матрицы Гессиана \mathbf{H} является число обусловленности

$$\varkappa(\mathbf{H}) = \frac{\lambda_{\max}(\mathbf{H})}{\lambda_{\min}(\mathbf{H})},$$

где $\lambda_{\max}(\mathbf{H}), \lambda_{\min}(\mathbf{H})$ являются максимальным и минимальным собственными значениями \mathbf{H} . Большое число обусловленности $\varkappa(\mathbf{H})$ приводит к нестабильности процесса оптимизации. Предложенный метод уменьшает размер матрицы Гессиана \mathbf{H} . Согласно экспериментам, приведенным в разделе 4.4 предлагаемый метод приводит к меньшему числу обусловленности $\varkappa(\mathbf{H})$.

Размер шага метода Ньютона может быть чрезмерно большим. Для контроля размера шага обновлений добавим параметр η в правило обновления (4.4)

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \eta\Delta\boldsymbol{\theta}^{k-1}, \quad \eta \in [0, 1].$$

Для выбора соответствующего размера шага η используется правило Армихо [98]. Выбирается максимальное η так, чтобы выполнялось условие

$$S(\boldsymbol{\theta}^{k-1} + \eta\Delta\boldsymbol{\theta}^{k-1}) < S(\boldsymbol{\theta}^{k-1}) + \gamma\eta\nabla S^\top(\boldsymbol{\theta}^{k-1})\boldsymbol{\theta}^{k-1}, \quad \gamma \in [0, 0.5].$$

Теорема 4. Пусть модель $f(\mathbf{x}, \boldsymbol{\theta})$ близка к линейной в окрестности точки $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$

$$\mathbf{f}(\mathbf{X}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \mathbf{f}(\mathbf{X}, \boldsymbol{\theta}) + \mathbf{J} \cdot \Delta\boldsymbol{\theta}, \quad (4.7)$$

где $\mathbf{J} \in \mathbb{R}^{m \times p}$ является матрицей Якоби

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1, \boldsymbol{\theta})}{\partial \theta_1} & \dots & \frac{\partial f(\mathbf{x}_1, \boldsymbol{\theta})}{\partial \theta_p} \\ \dots & \dots & \dots \\ \frac{\partial f(\mathbf{x}_m, \boldsymbol{\theta})}{\partial \theta_1} & \dots & \frac{\partial f(\mathbf{x}_m, \boldsymbol{\theta})}{\partial \theta_p} \end{pmatrix}.$$

Тогда вектор обновления $\Delta\boldsymbol{\theta}$ для функции ошибки (4.2) является решением задачи линейной регрессии

$$\|\mathbf{e} - \mathbf{F}\Delta\boldsymbol{\theta}\|_2^2 \rightarrow \min_{\Delta\boldsymbol{\theta} \in \mathbb{R}^p}, \quad (4.8)$$

где $\mathbf{e} = \mathbf{f} - \mathbf{y}$ и $\mathbf{F} = \mathbf{J}$.

Доказательство. В соответствии предположением (4.7) градиент $\nabla S(\boldsymbol{\theta})$ и матрица Гессиана \mathbf{H} имеют вид

$$\nabla S(\boldsymbol{\theta}) = \mathbf{J}^\top(\mathbf{y} - \mathbf{f}), \quad \mathbf{H} = \mathbf{J}^\top \mathbf{J}. \quad (4.9)$$

Тогда шаг метода Ньютона (4.6) и правило обновления (4.4) принимают вид

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \Delta\boldsymbol{\theta}^{k-1} = \boldsymbol{\theta}^{k-1} + (\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top(\mathbf{f} - \mathbf{y}).$$

Таким образом, согласно теореме Гаусса-Маркова, вектор обновления $\Delta\boldsymbol{\theta}$ является решением задачи регрессии (4.8). \square

В качестве нелинейной модели рассматривается модель двухслойной нейронной сети. В этом случае модель $f(\mathbf{x}, \boldsymbol{\theta})$ принимает вид:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \mathbf{W}_1) \mathbf{w}_2.$$

Здесь $\mathbf{W}_1 \in \mathbb{R}^{m \times h}$ — матрица параметров, которые соединяют исходные признаки с h скрытыми нейронами. Функция нелинейности $\sigma(\cdot)$ применяется поэлементно. Параметры $\mathbf{w}_2 \in \mathbb{R}^{h \times 1}$ соединяют скрытые нейроны с выходом. Вектор параметров модели $\boldsymbol{\theta}$ представляет собой объединение векторизованных матриц $\mathbf{W}_1, \mathbf{w}_2$.

Теорема 5. Рассмотрим модель логистической регрессии вида $f(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^\top \boldsymbol{\theta})$ с сигмоидной функцией активации $\sigma(\cdot)$. Вектор обновлений $\Delta\boldsymbol{\theta}$ для функции ошибки (4.3) является решением задачи линейной регрессии

$$\|\mathbf{e} - \mathbf{F}\Delta\boldsymbol{\theta}\|_2^2 \rightarrow \min_{\Delta\boldsymbol{\theta} \in \mathbb{R}^p}, \quad (4.10)$$

где $\mathbf{e} = \mathbf{R}^{-1/2}(\mathbf{y} - \mathbf{f})$ и $\mathbf{F} = \mathbf{R}^{1/2}\mathbf{X}$.

Доказательство. Градиент и Гессиан функции ошибки (4.3) равны

$$\nabla S(\boldsymbol{\theta}) = \mathbf{X}^\top (\mathbf{f} - \mathbf{y}), \quad \mathbf{H} = \mathbf{X}^\top \mathbf{R} \mathbf{X}, \quad (4.11)$$

где \mathbf{R} — это диагональная матрица с диагональными элементами $f(\mathbf{x}_i, \boldsymbol{\theta}) \cdot (1 - f(\mathbf{x}_i, \boldsymbol{\theta}))$.

Правило обновления (4.4) в этом случае принимает вид

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + (\mathbf{X}^\top \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{y} - \mathbf{f}).$$

Таким образом, согласно теореме Гаусса-Маркова, вектор обновления $\Delta\boldsymbol{\theta}$ является решением задачи регрессии (4.8). \square

Данный алгоритм известен как итеративный алгоритм взвешенных наименьших квадратов (IRLS) [99].

4.3 Метод Ньютона с выбором параметров с помощью квадратичного программирования

Предлагается адаптация метода QPFS для решения задач (4.8) и (4.10). Матрица парных взаимодействий \mathbf{Q} и вектор релевантностей \mathbf{b} имеют вид

$$\mathbf{Q} = \text{Sim}(\mathbf{F}), \quad \mathbf{b} = \text{Rel}(\mathbf{F}, \mathbf{e}).$$

Утверждение 13. В оптимальной точке $\boldsymbol{\theta}^*$ вектор релевантностей $\mathbf{b} = \text{Rel}(\mathbf{F}, \mathbf{e})$ равен нулю.

Доказательство. Выборочный коэффициент корреляции равен нулю для ортогональных векторов. Покажем, что в оптимальной точке $\boldsymbol{\theta}^*$ вектор \mathbf{e} ортогонален столбцам матрицы \mathbf{F} . Условие оптимальности первого порядка гарантирует это свойство для модели нелинейной регрессии

$$\mathbf{F}^\top \mathbf{e} = \mathbf{J}^\top (\mathbf{f} - \mathbf{y}) = -\nabla S(\boldsymbol{\theta}^*) = \mathbf{0},$$

и для модели логистической регрессии

$$\mathbf{F}^\top \mathbf{e} = \mathbf{X} \mathbf{R}^{-1/2} \mathbf{R}^{1/2} (\mathbf{y} - \mathbf{f}) = \mathbf{X}^\top (\mathbf{y} - \mathbf{f}) = \nabla S(\boldsymbol{\theta}^*) = \mathbf{0}.$$

□

Данное утверждение используется в качестве индикатора активности параметра модели в определении 11. Псевдокод предлагаемого алгоритма приведён в алгоритме 2.

Algorithm 2 QPFS + Ньютон алгоритм

Вход: ε — допустимое отклонение;

τ — пороговое значение;

γ — параметр правила Армихо.

Выход: $\boldsymbol{\theta}^*$;

инициализировать $\boldsymbol{\theta}^0$;

$k := 1$;

повторять

вычислить \mathbf{e} и \mathbf{F} для (4.8) или (4.10) ;

$\mathbf{Q} := \text{Sim}(\mathbf{F})$, $\mathbf{b} := \text{Rel}(\mathbf{F}, \mathbf{e})$, $\alpha = \frac{\overline{\mathbf{Q}}}{\overline{\mathbf{Q}} + \overline{\mathbf{b}}}$;

$\mathbf{a} := \arg \min_{\mathbf{a} \geq 0, \|\mathbf{a}\|_1=1} \mathbf{a}^\top \mathbf{Q} \mathbf{a} - \alpha \cdot \mathbf{b}^\top \mathbf{a}$;

$\mathcal{A} := \{j : a_j = 1\}$;

вычислить $\nabla S(\boldsymbol{\theta}^{k-1})$, \mathbf{H} для (4.9) или (4.11);

$\Delta \boldsymbol{\theta}^{k-1} = -\mathbf{H}^{-1} \nabla S(\boldsymbol{\theta}^{k-1})$;

$\eta := \text{ArmijoRule}(\boldsymbol{\theta}^{k-1}, \gamma)$;

$\boldsymbol{\theta}_{\mathcal{A}}^k = \boldsymbol{\theta}_{\mathcal{A}}^{k-1} + \eta \Delta \boldsymbol{\theta}_{\mathcal{A}}^{k-1}$;

$k := k + 1$;

пока $\frac{\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1}\|}{\|\boldsymbol{\theta}^k\|} < \varepsilon$

4.4 Анализ значимостей параметров нелинейных моделей

Целью вычислительного эксперимента является исследование свойств предложенного метода и сравнение его с другими методами.

Исследована зависимость параметров метода QPFS для задачи нелинейной регрессии (4.8) и задачи логистической регрессии (4.10). Предположим, что вектор параметров $\boldsymbol{\theta}^0$ лежит вблизи оптимального вектора параметров $\boldsymbol{\theta}^*$. Рассмотрим отрезок

$$\boldsymbol{\theta}_\beta = \beta \boldsymbol{\theta}^* + (1 - \beta) \boldsymbol{\theta}^0; \quad \beta \in [0, 1].$$

Сгенерируем синтетический набор данных с 300 объектами и 7 признаками для задачи логистической регрессии. Ландшафт функции ошибки (4.3) на сетке двух случайно выбранных параметров показан на Рис. 4.1. Поверхность функции ошибки выпуклая с вытянутыми линиями уровня вдоль некоторых параметров модели. Добавим случайный шум к оптимальным параметрам $\boldsymbol{\theta}^*$, чтобы получить точку $\boldsymbol{\theta}^0$. Поведение вектора \mathbf{b} на отрезке между $\boldsymbol{\theta}^0$ и $\boldsymbol{\theta}^*$ показано на Рис. 4.2. Компоненты \mathbf{b} начинают резко уменьшаться по мере приближения к оптимальной точке $\boldsymbol{\theta}^*$.

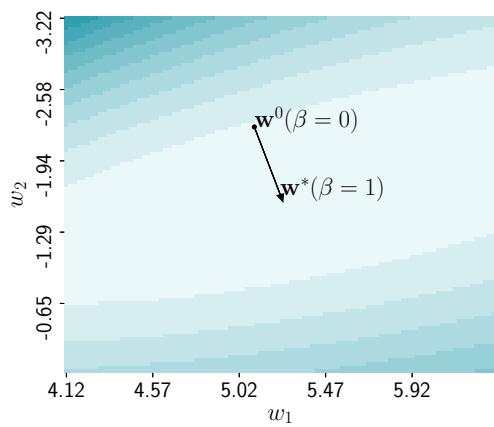


Рис. 4.1: Поверхность функции ошибки для логистической регрессии

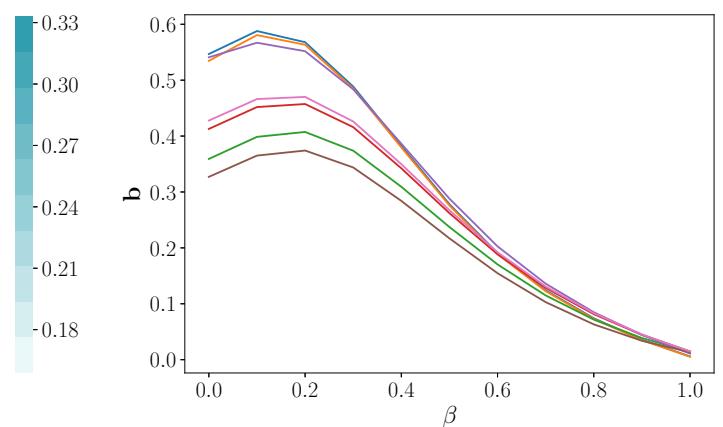


Рис. 4.2: Релевантность параметров для логистической регрессии

Для модели нелинейной регрессии используется классический набор данных Boston Housing с 506 объектами и 13 признаками. Для простоты нейронная сеть

содержит два скрытых нейрона. Ландшафт функции ошибок для модели нейронной сети является более сложным. Функция ошибки не является выпуклой и содержит множество локальных минимумов. Двумерный ландшафт функции ошибок для этого набора данных показан на Рис. 4.3. Сетка строится для двух случайных параметров из матрицы \mathbf{W}_1 . Аналогично на Рис. 4.4 показано, как изменяется вектор \mathbf{b} при движении от точки $\boldsymbol{\theta}^0$ до точки $\boldsymbol{\theta}^*$. Компоненты вектора \mathbf{b} становятся близки к нулю вблизи оптимума. При достижении оптимального значения различные параметры влияют на остатки модели \mathbf{e} .

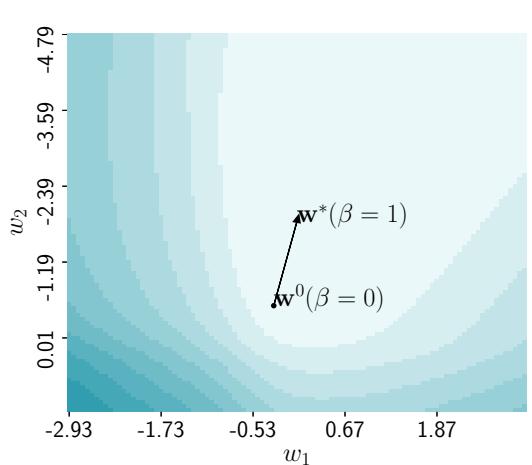


Рис. 4.3: Поверхность функции ошибки для нейронной сети

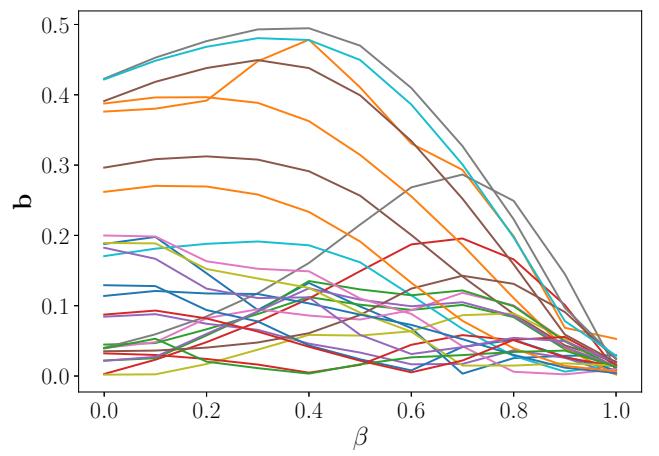


Рис. 4.4: Релевантность параметров первого слоя для модели нейронной сети

На Рис. 4.5 показан процесс оптимизации для предложенного метода в случае логистической регрессии с двумя параметрами модели. Даже для двумерной задачи решение метода Ньютона нестабильно и число обусловленности $\kappa(\mathbf{H})$ матрицы Гессиана \mathbf{H} может быть чрезвычайно большим. На каждом шаге алгоритма метод QPFS выбирает активные параметры для оптимизации. В данном примере предложенный метод выбирает и обновляет только один параметр на каждой итерации на первых шагах. Это делает метод более устойчивым.

На Рис. 4.6 показаны наборы активных параметров на итерациях для набора данных Boston Housing и нейронной сети с двумя скрытыми нейронами. Темные

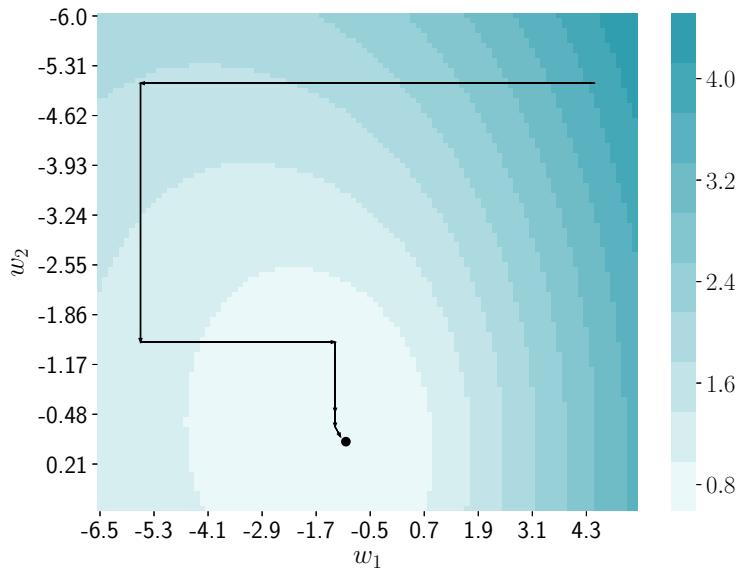


Рис. 4.5: Оптимизационный процесс предложенного метода QPFS+Ньютон для модели логистической регрессии

ячейки соответствуют активным параметрам, которые мы оптимизируем.

В рассмотренных примерах число обусловленности $\kappa(\mathbf{H})$ для метода Ньютона на некоторых итерациях было чрезвычайно большим. Выбор активных параметров позволил значительно сократить число обусловленности.

Приведём сравнение предложенного метода с существующими методами, а именно градиентным спуском (GD), моментом Нестерова [91], Adam [93] и оригинальным методом Ньютона. Проведены эксперименты для моделей нелинейной и логистической регрессий. Наборы данных были выбраны из репозитория UCI [100]. Результаты показаны в таблицах 4.1 и 4.2. Для каждого набора данных две строки таблиц содержат ошибки для тренировочной (первая строка) и тестовой (вторая строка) выборок. В таблице 4.1 приведена квадратичная ошибка, в таблице 4.2 — кросс-энтропия. Чтобы найти среднюю ошибку и ее стандартное отклонение использовалась процедура кросс валидации с разбиением на 5 фолдов. Предложенный метод показывает меньшую ошибку на трех из четырех наборов данных для нелинейной регрессии и на двух из трех наборов данных для логистической регрессии.

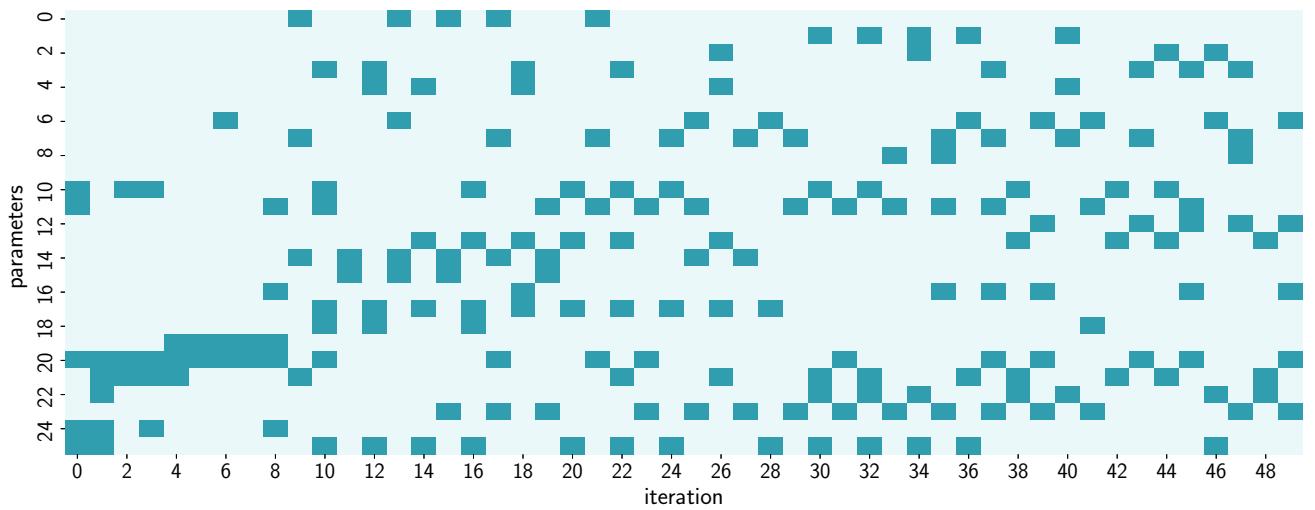


Рис. 4.6: Множество активных параметров на протяжении оптимизационного процесса

Таблица 4.1: Средняя квадратичная ошибка рассматриваемых алгоритмов оптимизации для модели нелинейной регрессии

Выборка	m n	GD	Нестеров	ADAM	Ньютон	QPFS+Ньютон
Boston House Prices	506	27.2 ± 4.6	46.0 ± 11.0	35.4 ± 2.5	22.1 ± 15.2	20.9 ± 10.4
Communities and Crime	1994 99	48.0 ± 6.4 47.5 ± 6.5	31.4 ± 2.8 32.9 ± 4.3	23.3 ± 3.7 28.1 ± 4.5	18.3 ± 3.4 28.8 ± 3.6	26.7 ± 3.1 $\mathbf{28.4 \pm 3.0}$
Forest Fires	517 10	18.9 ± 0.4 $\mathbf{20.0 \pm 2.1}$	1.83 ± 0.4 20.2 ± 2.2	1.81 ± 0.6 $\mathbf{20.0 \pm 2.0}$	17.7 ± 0.4 20.6 ± 1.4	17.9 ± 0.4 20.2 ± 2.2
Residential Building	372 103	51.6 ± 17.7 53.7 ± 13.9	32.6 ± 19.5 34.1 ± 13.6	30.0 ± 24.8 34.1 ± 19.4	35.5 ± 24.7 35.0 ± 15.6	30.3 ± 10.7 $\mathbf{30.9 \pm 5.3}$

Таблица 4.2: Среднее значение кросс-энтропии рассматриваемых алгоритмов оптимизации для модели логистической регрессии

Выборка	$\frac{m}{n}$	GD	Нестеров	ADAM	Ньютон	QPFS+Ньютон
Breast Cancer	569 30	0.6 ± 0.1 0.9 ± 0.2	0.4 ± 0.1 1.0 ± 0.7	0.8 ± 0.2 1.2 ± 0.2	0.3 ± 0.1 1.0 ± 0.2	0.2 ± 0.1 1.1 ± 0.3
Cardiotocography	2126 21	11.5 ± 4.7 11.6 ± 5.8	11.5 ± 4.7 11.5 ± 5.7	8.8 ± 4.4 9.0 ± 2.6	11.5 ± 5.7 11.5 ± 4.7	7.7 ± 4.2 7.7 ± 4.7
Climate Model Simulation Crashes	540 18	1.2 ± 0.1 1.4 ± 2.0	1.0 ± 0.2 1.3 ± 0.7	1.5 ± 0.2 1.8 ± 0.3	1.0 ± 0.5 1.2 ± 0.5	0.8 ± 0.3 1.1 ± 0.4

Глава 5

Метрические методы анализа временных рядов

При использовании в качестве функции ошибки квадратичной ошибки предполагается, что целевое пространство является евклидовым. Данное предположение не всегда является адекватным. В данной главе ставится задача метрического обучения как поиск оптимальной метрики в целевом пространстве. Рассматриваются задачи кластеризации и классификации множества временных рядов.

5.1 Метрическое обучение в задачах кластеризации временных рядов

Задана исходная матрица $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times n}$. Требуется выявить кластерную структуру данных и разбить пространство исходной переменной \mathbb{X} на множество непересекающихся кластеров $\mathbb{Y} = \{1, \dots, K\}$, т. е. построить отображение $f : \mathbb{X} \rightarrow \mathbb{Y}$. Обозначим $y_i = f(\mathbf{x}_i)$, $y_i \in \mathbb{Y}$ — метка кластера вектора \mathbf{x}_i . Необходимо выбрать метки кластеров $\{y_i\}_{i=1}^m$ таким образом, чтобы расстояния между кластерами были максимальными.

Определение 12. Центроидом класса $e \in \mathbb{Y}$ набора векторов $\mathcal{E} = \{\mathbf{x}_i : i = 1, \dots, m, y_i = e\}$ по расстоянию ρ назовем вектор $\mathbf{c}_e \in \mathbb{R}^n$, такой что

$$\mathbf{c}_e = \arg \min_{\mathbf{c} \in \mathbb{R}^n} \sum_{\mathbf{x}_i \in \mathcal{E}} \rho(\mathbf{x}_i, \mathbf{c}). \quad (5.1)$$

В случае использования в качестве расстояния ρ евклидовой метрики формула (5.1) принимает вид

$$\mathbf{c} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i, \quad \mathbf{c}_e = \frac{\sum_{i=1}^m [y_i = e] \mathbf{x}_i}{\sum_{i=1}^m [y_i = e]}. \quad (5.2)$$

Здесь вектор \mathbf{c} является центроидом всех строк исходной матрицы \mathbf{X} .

Введем в пространстве исходной переменной \mathbb{X} расстояние Махalanобиса

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \quad (5.3)$$

где матрица трансформаций $\mathbf{A} \in \mathbb{R}^{n \times n}$ является симметричной и неотрицательно определенной ($\mathbf{A}^\top = \mathbf{A}$, $\mathbf{A} \succeq \mathbf{0}$). Зададим в качестве матрицы трансформации матрицу выборочной ковариации

$$\mathbf{A} = \frac{1}{m} \sum_{i=1}^m (\mathbf{x}_i - \mathbf{c})(\mathbf{x}_i - \mathbf{c})^\top. \quad (5.4)$$

Функцией ошибки кластеризации назовем межкластерное расстояние:

$$\mathcal{L}(\{\mathbf{c}_j\}_{j=1}^K, \mathbf{X}, \mathbf{y}) = - \sum_{j=1}^K m_j d_{\mathbf{A}}^2(\mathbf{c}_j, \mathbf{c}), \quad (5.5)$$

где $m_j = \sum_{i=1}^m [y_i = j]$ — число векторов в кластере j .

Поставим задачу кластеризации как задачу минимизации функции ошибки (5.5)

$$\mathcal{L}(\{\mathbf{c}_j\}_{j=1}^K, \mathbf{X}, \mathbf{y}) \rightarrow \min_{\mathbf{c}_j \in \mathbb{R}^n}. \quad (5.6)$$

Для решения этой задачи предлагается применить метод метрического обучения к матрице трансформации \mathbf{A} . Найдем такую матрицу \mathbf{A} , для которой функционал качества принимает максимальное значение:

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathbb{R}^{n \times n}} S(\{\mathbf{c}_j^*\}_{j=1}^K, \mathbf{X}, \mathbf{y}), \quad (5.7)$$

где $\{\mathbf{c}_j^*\}_{j=1}^K$ — решение задачи кластеризации (5.6).

5.2 Алгоритм адаптивного метрического обучения

Для решения задач (5.6), (5.7) используется алгоритм адаптивного метрического обучения. Предлагается понизить размерность пространства исходной переменной \mathbb{X} с помощью линейного преобразования $\mathbf{P} \in \mathbb{R}^{l \times n}$, $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$, где новая размерность $l < n$

$$\mathbf{t}_i = \mathbf{P} \mathbf{x}_i \in \mathbb{R}^l, \quad i = 1, \dots, m.$$

Центроид $\hat{\mathbf{c}}$ множества векторов $\{\mathbf{t}_i\}_{i=1}^m$ вычисляется по формуле (5.2). Расстояния между векторами вычисляются по формуле (5.3), где в качестве матрицы $\hat{\mathbf{A}}$

используется матрица ковариаций (5.4) множества векторов $\{\hat{\mathbf{t}}_i\}_{i=1}^m$

$$\hat{\mathbf{A}} = \frac{1}{m} \sum_{i=1}^m (\mathbf{t}_i - \hat{\mathbf{c}})(\mathbf{t}_i - \hat{\mathbf{c}})^T = \frac{1}{m} \sum_{i=1}^m \mathbf{P}(\mathbf{x}_i - \mathbf{c})(\mathbf{x}_i - \mathbf{c})^T \mathbf{P}^T = \mathbf{P} \mathbf{A} \mathbf{P}^T.$$

Определение 13. *Взвешенной индикаторной матрицей* назовем матрицу $\mathbf{N} \in \mathbb{R}^{m \times K}$, элементы которой равны:

$$n_{ij} = \begin{cases} \frac{1}{\sqrt{m_j}}, & \text{если } f(\mathbf{x}_i) = y_j; \\ 0, & \text{если } f(\mathbf{x}_i) \neq y_j. \end{cases}$$

В работе [101] показано, что с использованием данных обозначений задача кластеризации (5.6) и задача метрического обучения (5.7) сводятся к общей задаче минимизации функции ошибки

$$\begin{aligned} \mathcal{L} &= -\frac{1}{m} \text{trace}(\mathbf{N}^T \mathbf{X}^T \mathbf{P}^T \hat{\mathbf{A}}^{-1} \mathbf{P} \mathbf{X} \mathbf{N}) = \\ &= -\frac{1}{m} \text{trace}(\mathbf{N}^T \mathbf{X}^T \mathbf{P}^T (\mathbf{P} \mathbf{A} \mathbf{P}^T)^{-1} \mathbf{P} \mathbf{X} \mathbf{N}) \rightarrow \min_{\mathbf{P}, \mathbf{N}}. \quad (5.8) \end{aligned}$$

Для решения задачи (5.8) используется ЕМ алгоритм. На каждом шаге итеративно вычисляются текущие оптимальные значения матриц \mathbf{P} и \mathbf{N} . На E -шаге необходимо найти матрицу \mathbf{N} , которая является решением оптимизационной задачи (5.8) при фиксированной матрице \mathbf{P} . В качестве начального приближения получим взвешенную индикаторную матрицу \mathbf{N} с помощью алгоритма кластеризации k -средних с евклидовой метрикой. На M -шаге производится нахождение оптимального значения матрицы \mathbf{P} при фиксированной матрице \mathbf{N} . Алгоритм завершается при стабилизации функционала \mathcal{L} на последовательности итераций.

Алгоритм k -средних. В данной работе базовым алгоритмом для сравнения является алгоритм k -средних. На первом шаге алгоритм выбирает из матрицы \mathbf{X} случайным образом r строк $\{\mathbf{c}_j\}_{j=1}^K$ — начальные центроиды кластеров. Для каждого вектора \mathbf{x}_i вычисляется расстояние (5.3) до каждого центроида кластера \mathbf{c}_j с единичной матрицей трансформаций \mathbf{A} . Вектор \mathbf{x}_i относится к

кластеру, расстояние до которого оказалось наименьшим. Далее производится вычисление новых центроидов кластеров по формуле (5.2). Алгоритм завершается, если значения центроидов кластеров стабилизируются.

Оптимизация матрицы \mathbf{P} с фиксированной матрицей \mathbf{N} . Для любых двух квадратных матриц \mathbf{A} и \mathbf{B} справедливо $\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA})$. Данное свойство позволяет переформулировать задачу (5.8) следующим образом:

$$\mathcal{L} = -\frac{1}{m} \text{trace}(\mathbf{N}^T \mathbf{X}^T \mathbf{P}^T (\mathbf{P} \mathbf{A} \mathbf{P}^T)^{-1} \mathbf{P} \mathbf{X} \mathbf{N}) = -\frac{1}{m} \text{trace}((\mathbf{P} \mathbf{A} \mathbf{P}^T)^{-1} \mathbf{P} \mathbf{X} \mathbf{N} \mathbf{N}^T \mathbf{X}^T \mathbf{P}^T).$$

Утверждение 14. Обозначим $\mathbf{B} = \mathbf{X} \mathbf{N} \mathbf{N}^T \mathbf{X}^T$. Обозначим через $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_r]^T$ матрицу, состоящую из r собственных векторов матрицы $\mathbf{A}^{-1} \mathbf{B}$, отвечающих наибольшим собственным значениям. Тогда решением (5.8) является ортогональная матрица, полученная QR-разложением матрицы \mathbf{P}^T .

Доказательство. Функция ошибки \mathcal{L} (5.5) зависит только от матрицы \mathbf{P} . Обозначим

$$s(\mathbf{P}) = \text{trace}((\mathbf{P} \mathbf{A} \mathbf{P}^T)^{-1} \mathbf{P} \mathbf{B} \mathbf{P}^T).$$

На данном шаге задача (5.8) принимает вид:

$$\mathbf{P}^* = \arg \max_{\mathbf{P} \in \mathbb{R}^{l \times n}} s(\mathbf{P}); \quad (5.9)$$

$$\mathbf{P} \mathbf{P}^T = \mathbf{I}. \quad (5.10)$$

Ранг произведения матриц не превосходит рангов сомножителей, поэтому ранг матрицы \mathbf{B} не превосходит K . Решением (5.9) является матрица $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_K]^T$, состоящая из K собственных векторов матрицы $\mathbf{A}^{-1} \mathbf{B}$, отвечающих наибольшим собственным значениям. Таким образом, размерность нового пространства будет равна количеству кластеров K .

В общем случае матрица \mathbf{P} не является ортогональной. Заметим, что для любой невырожденной матрицы \mathbf{P} верно $s(\mathbf{P}) = s(\mathbf{MP})$. Для учета условия ортогональности (5.10) найдем QR-разложение матрицы \mathbf{P} . Тогда ортогональная матрица \mathbf{Q} является оптимальным значением \mathbf{P}^* . \square

Оптимизация матрицы \mathbf{N} с фиксированной матрицей \mathbf{P} . Обозначим $\hat{\mathbf{K}} = \frac{1}{m} \mathbf{X}^\top \mathbf{P}^\top \hat{\mathbf{A}}^{-1} \mathbf{P} \mathbf{X}$. В работе [102] показано, что тогда задача (5.8) эквивалентна задаче кластеризации k -средних с заданным ядром $\hat{\mathbf{K}}$.

При фиксированной матрице \mathbf{P} задача (5.8) принимает вид:

$$\text{trace}(\mathbf{N}^\top \hat{\mathbf{K}} \mathbf{N}) \rightarrow \max_{\mathbf{N} \in \mathbb{R}^{m \times r}}.$$

Матрица $\hat{\mathbf{K}}$ является симметричной и неотрицательно определенной, тем самым может быть выбрана в качестве ядра.

5.3 Задача метрического обучения с динамическим выравниванием временных рядов

Пусть исходная переменная $\mathbf{x} \in \mathbb{X}$ — временной ряд, $\mathbf{y} \in \mathbb{Y} = \{1, \dots, K\}$ — метка класса. Требуется построить точную, простую, устойчивую модель классификации $a : \mathbb{X} \rightarrow \mathbb{Y}$. Данную модель представим в виде суперпозиции

$$a(\mathbf{x}) = b \circ \mathbf{f} \circ G(\mathbf{x}, \{\mathbf{c}_e\}_{e=1}^K),$$

где G — процедура выравнивания временных рядов относительно центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$, \mathbf{f} — алгоритм метрического обучения, b — алгоритм многоклассовой классификации.

Выравнивание временных рядов. Для повышения качества и устойчивости алгоритма классификации предлагается провести выравнивание временных рядов каждого класса относительно центроида, введенного в определении 12.

Для нахождения центроида предлагается в качестве расстояния ρ между временными рядами использовать путь наименьшей стоимости [103, 104], найденный методом динамической трансформации времени. Псевдокод решения оптимизационной задачи (5.1) приведен в алгоритме 3. Общая процедура выравнивания имеет следующий вид:

- 1) построить множество центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$;

Algorithm 3 Нахождение центроида DBA(\mathbf{X}_e , n_iter)

Вход: \mathbf{X}_e — множество временных рядов, принадлежащих одному и тому же классу, n_iter — количество итераций алгоритма.

Выход: \mathbf{c} — центроид множества \mathbf{X}_e .

- 1: задать начальное приближение приближение центроида \mathbf{c} ;
- 2: **для** $i = 1, \dots, n_{\text{iter}}$
- 3: **для** $\mathbf{x} \in \mathbf{X}_e$
- 4: вычислить выравнивающий путь между \mathbf{c} и \mathbf{x}
 $\text{alignment}(\mathbf{x}) := \text{DTWalignment}(\mathbf{c}, \mathbf{x});$
- 5: объединить поэлементно множества индексов для каждого отсчета времени
 $\text{alignment} := \bigcup_{\mathbf{x} \in \mathbf{X}_e} \text{alignment}(\mathbf{x});$
- 6: $\mathbf{c} = \text{mean}(\text{alignment})$

DTWalignment(\mathbf{c}, \mathbf{x})

Вход: \mathbf{c}, \mathbf{x} — временные ряды.

Выход: alignment — выравнивающий путь. // каждый индекс временного ряда \mathbf{x} поставлен в однозначное соответствие индексу временного ряда \mathbf{c}

- 1: построить $n \times n$ -матрицу деформаций DTW
 $\text{cost} := \text{DTW}(\mathbf{c}, \mathbf{x});$
 - 2: вычислить выравнивающий путь по матрице деформаций
 $\text{alignment} := \text{DTWpath}(\text{cost});$
-

- 2) по множеству центроидов найти пути наименьшей стоимости между каждым временным рядом \mathbf{x}_i и центроидом его класса \mathbf{c}_{y_i} ;
- 3) по каждому пути восстановить выравненный временной ряд;
- 4) привести множества выравненных временных рядов к нулевому среднему и нормировать на дисперсию.

Результатом выравнивания должно стать множество выравненных временных рядов.

Метрическое обучение. Введем на множестве выравненных временных рядов расстояние Махalanобиса $d_{\mathbf{A}}$ (5.3). Представим матрицу трансформации \mathbf{A} в виде разложения $\mathbf{A}^{-1} = \mathbf{L}^T \mathbf{L}$. Матрица $\mathbf{L} \in \mathbb{R}^{p \times n}$ — матрица линейного преобразования, где p задает размерность преобразованного пространства. Если параметр $p < n$, то происходит снижение размерности признакового пространства.

Расстояние $d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)$ есть евклидово расстояние между $\mathbf{L}\mathbf{x}_i$ и $\mathbf{L}\mathbf{x}_j$:

$$\begin{aligned} d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{L}^T \mathbf{L} (\mathbf{x}_i - \mathbf{x}_j)} = \\ &= \sqrt{(\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))^T (\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))} = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2. \end{aligned}$$

В качестве алгоритма метрического обучения в данной работе был выбран алгоритм LMNN [20]. Данный алгоритм сочетает в себе идеи метода k ближайших соседей. Первая идея заключается в минимизации расстояний между k ближайшими векторами, находящимися в одном классе. Запишем функционал качества в виде

$$Q_1(\mathbf{L}) = \sum_{j \sim i} \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \rightarrow \min_{\mathbf{L}},$$

где запись $j \sim i$ означает, что \mathbf{x}_j является одним из k ближайших соседей для \mathbf{x}_i . Вторая идея состоит в максимизации расстояния между каждым вектором и его векторами-нарушителями.

Определение 14. *Вектором-нарушителем* для \mathbf{x}_i назовем вектор \mathbf{x}_l такой, что

$$\|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2 \leq \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 + 1, \quad \text{где } j \sim i. \quad (5.11)$$

Таким образом, необходимо минимизировать следующий функционал:

$$Q_2(\mathbf{L}) = \sum_{j \sim i} \sum_l [y_i \neq y_l] [1 + \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 - \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2]_+ \rightarrow \min_{\mathbf{L}}.$$

Положительная срезка позволяет штрафовать только те вектора, которые удовлетворяют условию (5.11).

Задача метрического обучения состоит в нахождении линейного преобразования $\mathbf{f}(\mathbf{x}) = \mathbf{L}\mathbf{x}$, то есть нахождении матрицы \mathbf{L} в виде решения оптимизационной задачи

$$Q(\mathbf{L}) = \mu Q_1(\mathbf{L}) + (1 - \mu)Q_2(\mathbf{L}) \rightarrow \min_{\mathbf{L}}, \quad (5.12)$$

где $\mu \in (0, 1)$ — весовой параметр, определяющий вклад каждого из функционалов. Задача (5.12) представляет собой задачу полуопределенного программирования [105] и может быть решена существующими оптимизационными пакетами.

На Рис. 5.1 показан принцип работы алгоритма метрического обучения LMNN по сравнению с базовым методом, использующим евклидову метрику, для случая двумерных данных. Алгоритм LMNN позволяет найти оптимальную матрицу трансформации \mathbf{A} , отдаляя вектора разных классов и притягивая вектора одного класса. В случае использования евклидовой метрики матрица трансформаций \mathbf{A} является единичной матрицей.

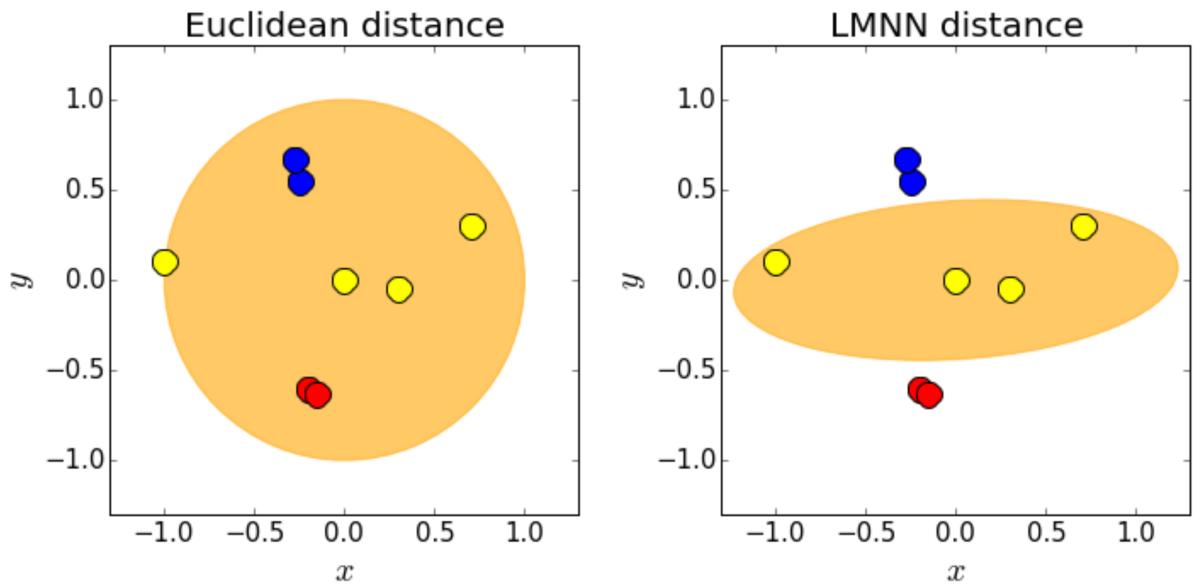


Рис. 5.1: Сравнение оптимальной метрики Махalanобиса алгоритма LMNN с евклидовой метрикой в двумерном случае

Классификация выравненных временных рядов в метрике Махalanобиса. Пусть $\mathbf{x} \in \mathbf{X}$ — неразмеченный временной ряд. Выравниваем вре-

менной ряд \mathbf{x} относительно всех центроидов классов

$$\hat{\mathbf{x}}_e = G(\mathbf{x}, \mathbf{c}_e), \quad \text{где } e \in \{1, \dots, K\}.$$

Отнесем временной ряд к классу, для которого минимально расстояние до соответствующего центроида. В качестве расстояния используем обученную метрику Махalanобиса с фиксированной матрицей \mathbf{A}

$$\hat{y} = \arg \min_{e \in \mathbb{Y}} d_{\mathbf{A}}(\hat{\mathbf{x}}_e, \mathbf{c}_e).$$

После нахождения оптимальных центроидов классов и нахождения оптимальной матрицы трансформаций процедура классификации заключается в измерении расстояния между найденными центроидами и новыми неразмеченными векторами.

Для оценки качества работы алгоритма будем вычислять ошибку классификации как долю неправильно классифицированных векторов тестовой выборки $\{\mathbf{x}_i, y_i\}_{i=1}^{\hat{m}}$:

$$\text{error} = \frac{1}{\hat{m}} \sum_{i=1}^{\hat{m}} [a(\mathbf{x}_i) \neq y_i].$$

5.4 Анализ метрического пространства для задачи кластеризации

В целях проверки работоспособности предложенного подхода проведен вычислительный эксперимент на модельных данных. Сгенерирована выборка точек, принадлежащих одному из двух классов, в двумерном пространстве. Каждая точка принадлежит многомерному нормальному распределению. На Рис. 5.2 показано истинное распределение точек, черным цветом выделены истинные центры классов и линии уровня функции распределения.

Применим к данной выборке базовый алгоритм k -средних. Результат кластеризации показан на Рис. 5.3, где черным цветом выделены найденные центры классов и линии уровня функции распределения, построенной по выборочной матрице ковариаций. Взяв за начальное приближение результаты работы алго-

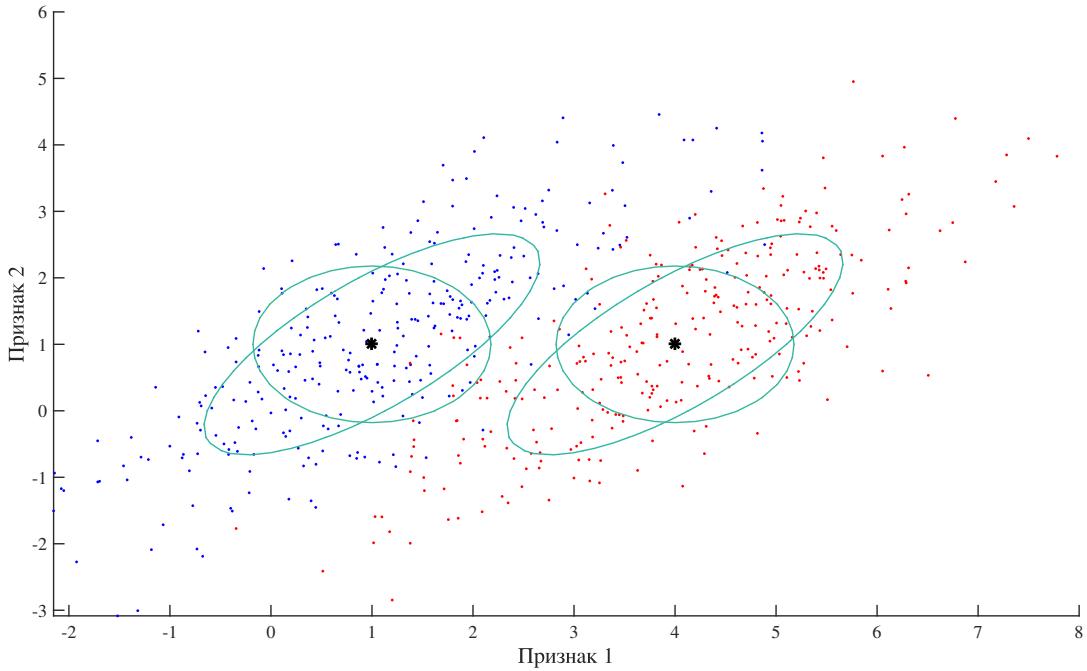


Рис. 5.2: Истинное распределение двумерных модельных данных

ритма k -средних, проведем кластеризацию с помощью алгоритма адаптивного метрического обучения. Результаты работы алгоритма продемонстрированы на Рис. 5.4.

На рисунках заметно улучшение результатов кластеризации. Измеренная точность кластеризации алгоритма k -средних составила 0,76, алгоритма адаптивного метрического обучения — 0,94, что говорит об эффективности данного подхода.

Таблица 5.1 показывает результаты вычислительного эксперимента на реальных данных. Алгоритм был применен к 5 выборкам, взятых из репозитория UCI [100]. Оценкой качества кластеризации служит число правильно кластеризованных векторов. При кластеризации векторов на более чем два класса возникает проблема соотнесения истинных классов с полученными кластерами. Данная проблема была формализована в виде задачи о назначениях и решена с помощью венгерского алгоритма. Вычислительный эксперимент на реальных данных показал увеличение точности кластеризации при использовании метри-

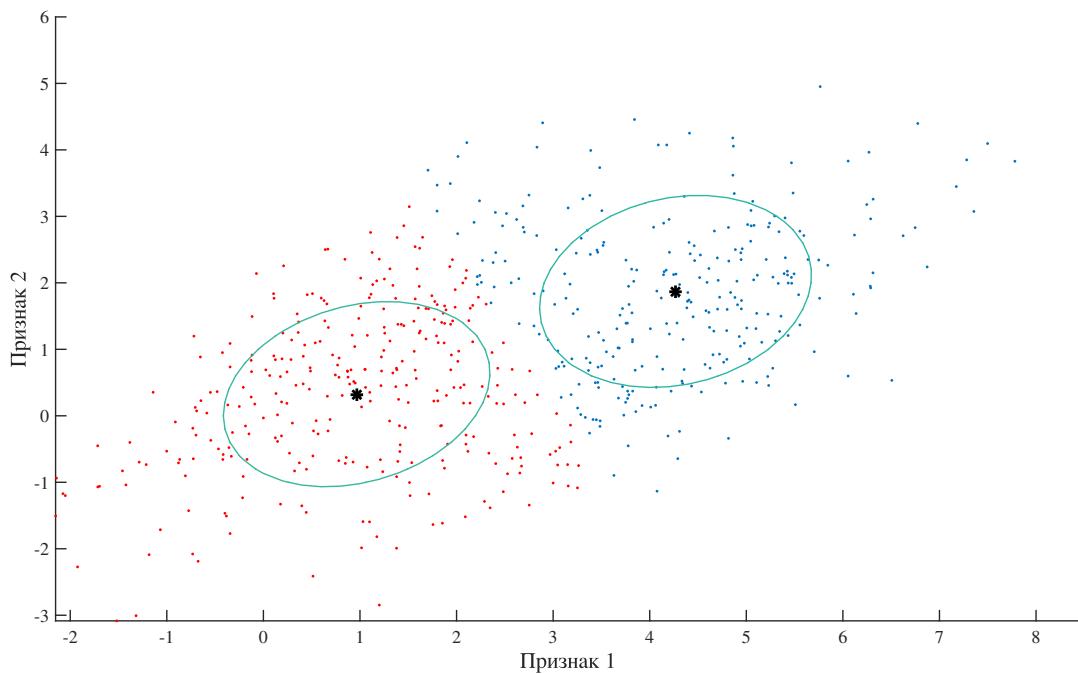


Рис. 5.3: Результат кластеризации модельных данных алгоритмом k -средних

ческого обучения.

Таблица 5.1: Результаты кластеризации на множестве датасетов для методов k -средних и AML

Выборка	Качество кластеризации	
	k -средних	AML
Letter Recognition	0,356	0,428
Optical Recognition of Handwritten Digits	0,758	0,790
Seeds	0,833	0,881
Image Segmentation	0,545	0,737
Breast Cancer Wisconsin	0,960	0,956

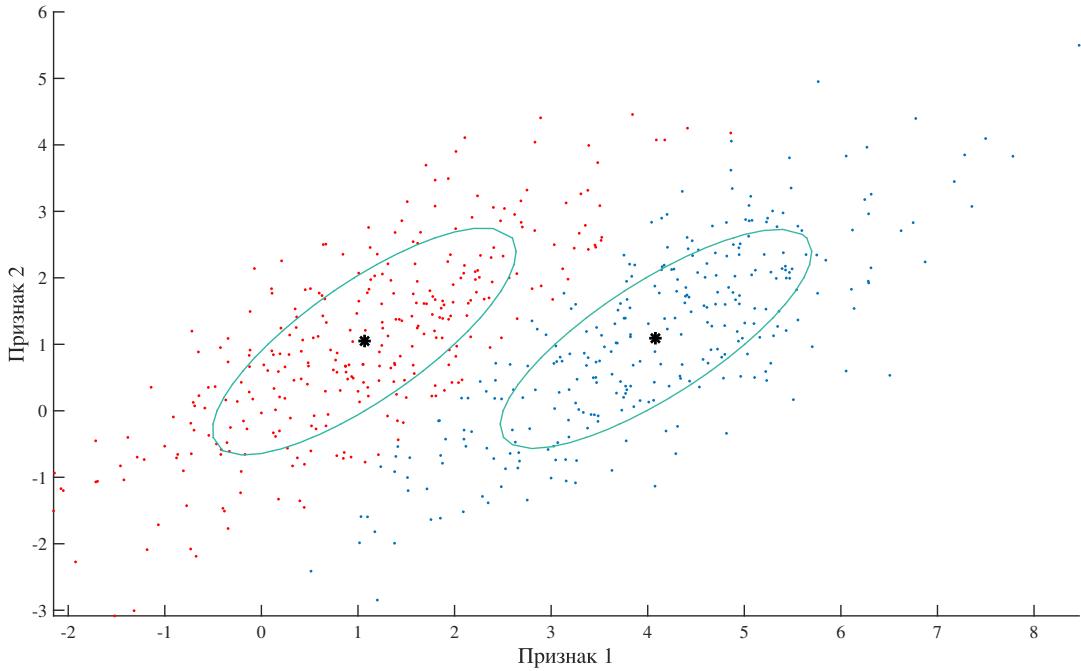


Рис. 5.4: Результат кластеризации модельных данных алгоритмом адаптивного метрического обучения

5.5 Анализ метрического пространства для задачи классификации временных рядов

Цель вычислительного эксперимента — проверить работоспособность предложенного подхода. Предполагается, что построенный алгоритм мультиклассовой классификации способен определить тип активности человека по форме сигнала акселерометра мобильного телефона.

Для проведения базового вычислительного эксперимента были подготовлены синтетические временные ряды, принадлежащие двум классам. Первый класс — синусы вида $\sin(x + b)$, где параметр b определяет сдвиг каждого временного ряда. Второй класс — пилюобразные функции с различными сдвигами по временной шкале. На каждый временной ряд был наложен нормальный шум. Число временных рядов каждого класса = 60. Длина каждого временного ряда $n = 50$.

Построенные центроиды классов проиллюстрированы на Рис. 5.5. Из рисун-

ка видно, что процедура корректно определяет сдвиги временных рядов.

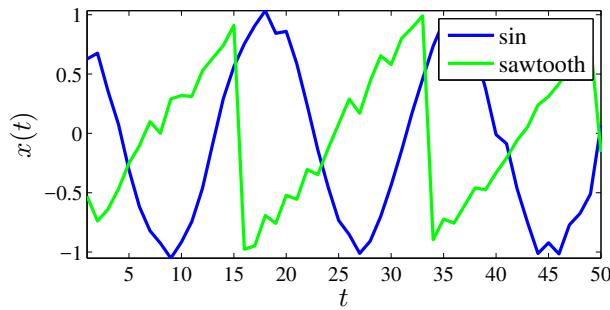


Рис. 5.5: Примеры центроидов синтетических временных рядов

Для того чтобы убедиться в целесообразности применения метрического обучения, данные временные ряды классифицировались в пространстве с евклидовой метрикой и в пространстве с метрикой Махalanобиса. Число ближайших соседей $k = 5$, размерность преобразованного пространства $p = 40$. Полученные ошибки классификации составили 27% для евклидовой метрики и 6% для метрики Махalanобиса.

Реальные данные [106] представляли собой временные ряды акселерометра мобильного телефона. Каждый из шести классов соответствовал определенной физической активности испытуемых. Для проведения вычислительного эксперимента было выбрано по 200 объектов каждого класса. Длина каждого временного ряда равнялась $n = 128$ отсчетам времени.

Построенные центроиды классов изображены на Рис. 5.6. Найденные центроиды обладают периодичностью, свойственной времененным рядам показаний активности человека. На Рис. 5.7 показаны примеры временных рядов каждого класса. Эти же временные ряды после процедуры выравнивания относительно построенных центроидов изображены на Рис. 5.8.

Ошибка классификации без использования метрического обучения составила 37,5%. Алгоритм LMNN позволяет настроить параметры: число ближайших соседей k , размерность преобразованного евклидова пространства p . Для выбора оптимальных параметров воспользуемся процедурой кросс-валидации. На

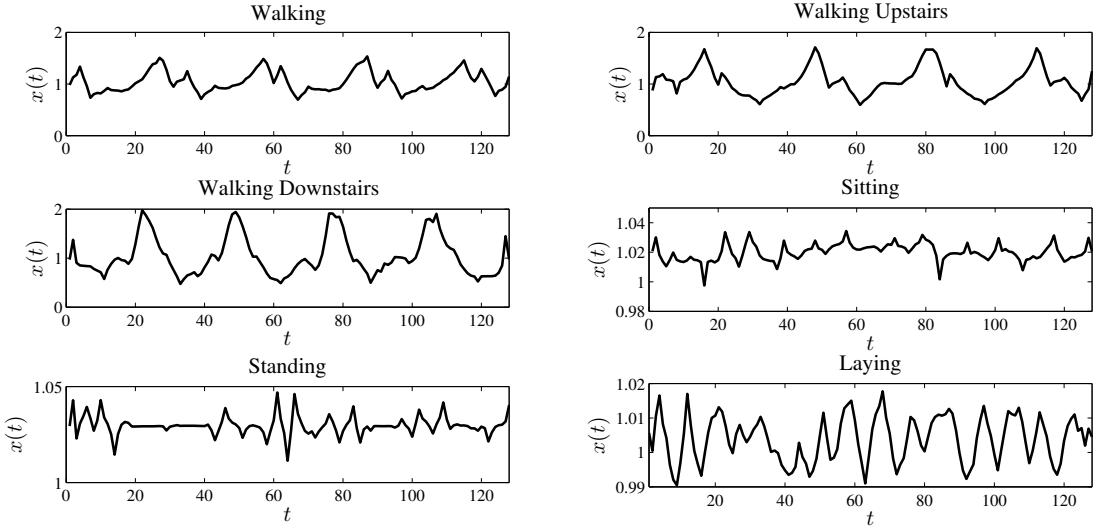


Рис. 5.6: Примеры центроидов временных рядов акселерометра

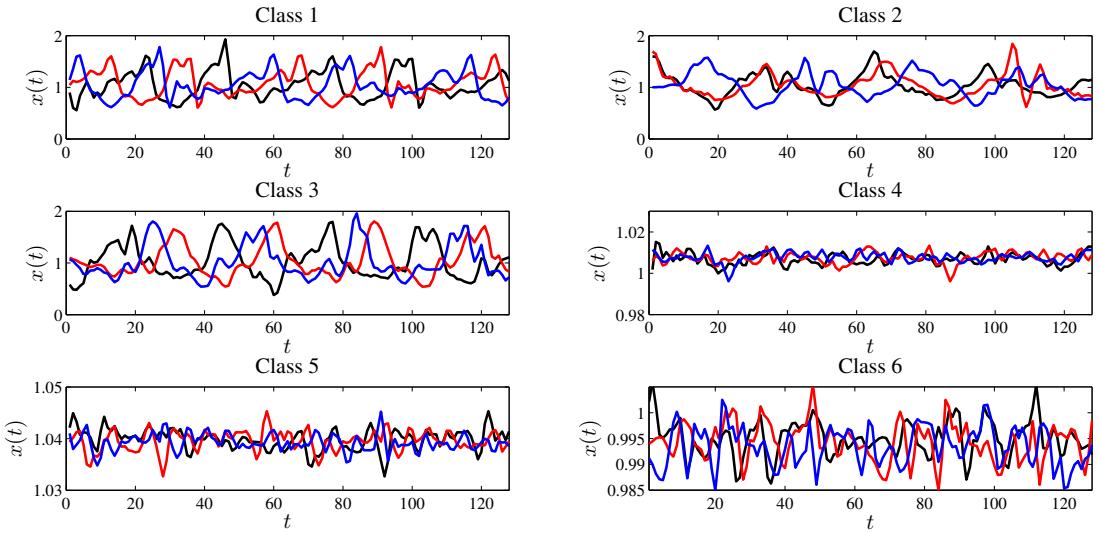


Рис. 5.7: Примеры временных рядов акселерометра

Рис. 5.9 цветом показана ошибка классификации алгоритма в зависимости от его параметров. На данной выборке алгоритм LMNN оказывается слабо чувствителен к числу ближайших соседей, и при уменьшении размерности пространства ошибка классификации растет.

Настроим алгоритм LMNN со следующими параметрами: число ближайших соседей $k = 30$, размерность выходного пространства $p = 128$. Ошибка

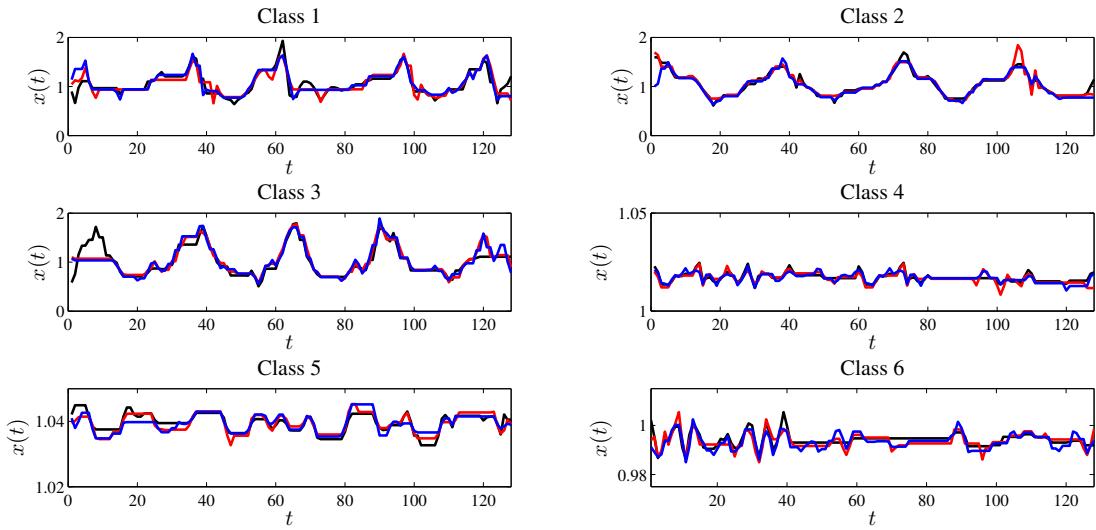


Рис. 5.8: Выравненные временные ряды акселерометра

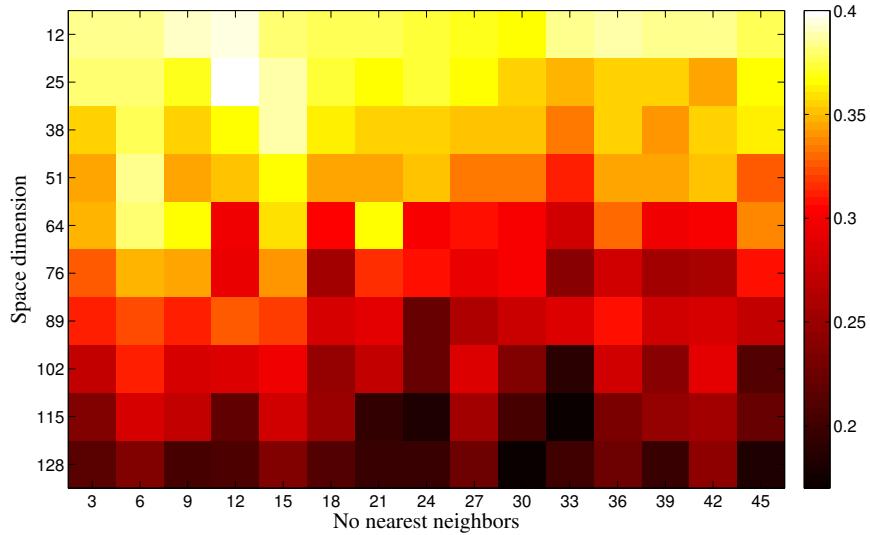


Рис. 5.9: Ошибка классификации метрического алгоритма в зависимости от размерности пространства и количества используемых ближайших соседей

классификации составила 17,25%, что вдвое меньше ошибки классификации с использованием евклидовой метрики.

В таблице 5.2 представлены матрицы несоответствий результатов классификации при использовании евклидовой метрики и метрики Махalanобиса. Столбцы соответствуют истинным меткам классов, строки — предсказанным меткам.

Таблица 5.2: Матрицы несоответствий для евклидовой метрики и метрики Махalanобиса, построенные для временных рядов акселерометра

(a) Евклидова метрика							(b) Метрика Махalanобиса						
	Истинные метки классов							Истинные метки классов					
	1	2	3	4	5	6		1	2	3	4	5	6
1	80	0	5	0	0	0		151	12	13	0	0	0
2	4	56	33	0	0	0		10	142	14	0	0	0
3	5	5	86	0	0	0		9	10	171	0	0	0
4	7	8	5	168	4	21		10	7	0	173	9	21
5	51	61	57	12	192	11		2	11	0	12	186	9
6	53	70	14	20	2	168		18	18	2	15	5	170

Диагональное преобладание матрицы несоответствий указывает на высокую предсказательную способность алгоритма.

В таблице 5.3 продемонстрировано увеличение точности классификации при использовании в качестве меры расстояния метрики Махalanобиса. Пересечение i -го столбца и j -й строки отвечает изменению доли объектов класса i , относенных к классу j . Положительное суммарное значение диагональных элементов таблицы соответствует увеличению качества классификации. Значительное улучшение предсказания происходит при классификации первых трех классов. Данные классы соответствуют следующим видам физической активности: ходьба, ходьба вверх, ходьба вниз.

Таблица 5.3: Прирост точности классификации при использовании адекватной оценки матрицы трансформаций

	Истинные метки классов					
	1	2	3	4	5	6
1	0,355	0,06	0,04	0	0	0
2	0,03	0,43	-0,095	0	0	0
3	0,02	0,025	0,425	0	0	0
4	0,015	-0,005	-0,025	0,025	0,025	0
5	-0,245	-0,25	-0,28	0	-0,03	-0,01
6	-0,175	-0,26	-0,06	-0,025	0,005	-0,01

Глава 6

Порождение признаков с помощью метамоделей

Исходное пространство сигналов в задачах декодирования, а также в задачах анализа временных рядов является крайне избыточным и неинформативным. Для извлечения информативных признаков в данной главе ставится задача порождения признакового пространства.

6.1 Постановка задачи порождения признакового пространства

Временные ряды акселерометра образуют множество \mathcal{S} сегментов $\mathbf{s} = [x_1, \dots, x_T]^T$ фиксированной длины T . Необходимо построить модель классификации $a : \mathbb{R}^T \rightarrow \mathbb{Y}$, которая будет ставить в соответствие каждому сегменту из множества \mathcal{S} метку класса из конечного множества $\mathbb{Y} = \{1, \dots, K\}$. Пусть $\{(\mathbf{s}_i, y_i)\}_{i=1}^m$ — исходная выборка, где $\mathbf{s}_i \in \mathcal{S}$ и $y_i = a(\mathbf{s}_i) \in \mathbb{Y}$.

В работе предлагается построить модель a в виде суперпозиции $a = f \circ g$.

Определение 15. *Порождающей функцией* будем называть функцию $g : \mathbb{R}^T \rightarrow \mathbb{X}$, отображающую исходные временные ряды \mathbf{s} из пространства \mathbb{R}^T в признаковое пространство $\mathbb{X} \subset \mathbb{R}^n$.

Имея порождающую функцию g , преобразуем исходную выборку в $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, где $\mathbf{x}_i = g(\mathbf{s}_i) \in \mathbb{X}$.

Модель классификации $f = f(\mathbf{x}, \boldsymbol{\theta})$ является параметрической функцией с вектором параметров $\boldsymbol{\theta}$. Оптимальные параметры $\boldsymbol{\theta}^*$ определяются оптимизацией функции ошибки классификации

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}, \boldsymbol{\mu}). \quad (6.1)$$

Вектор $\boldsymbol{\mu}$ является внешним параметром для заданной модели классификации. Примеры таких параметров и функций ошибки для различных моделей классификации приведены ниже.

Чтобы сравнить качество классификации с прошлыми результатами [107, 108], в качестве метрики качества используется точность классификации:

$$\text{accuracy} = \frac{1}{m} \sum_{i=1}^m [f(\mathbf{g}(\mathbf{s}_i), \boldsymbol{\theta}^*) = y_i]. \quad (6.2)$$

6.2 Модели порождения признакового пространства для временных рядов

Цель данной работы — провести сравнение различных подходов к генерации признаков. В этом разделе проводится анализ рассматриваемых методов.

Экспертные функции. В качестве базового подхода будем использовать экспертные функции как функции порождения признаков. Экспертные функции — это некоторые статистики g_j , где $g_j : \mathbb{R}^T \rightarrow \mathbb{R}$. Признаком описанием $\mathbf{g}(\mathbf{s})$ временного ряда \mathbf{s} являются значения заданных экспертных статистик

$$\mathbf{x} = \mathbf{g}(\mathbf{s}) = [g_1(\mathbf{s}), \dots, g_n(\mathbf{s})]^\top.$$

В работе [109] авторы предлагают использовать экспертные функции, приведенные в таблице 6.1. Такая процедура порождения признаков генерирует признаковое описание временного ряда $\mathbf{x} = \mathbf{g}(\mathbf{s}) \in \mathbb{R}^{40}$.

Таблица 6.1: Примеры экспертных порождающих функций

Описание	Формула
Mean	$\bar{s} = \frac{1}{T} \sum_{t=1}^T s_t$
Standard deviation	$\sqrt{\frac{1}{T} \sum_{t=1}^T (s_t - \bar{s})^2}$
Mean absolute deviation	$\frac{1}{T} \sum_{t=1}^T s_t - \bar{s} $
Distribution	Histogram values with 10 bins

Авторегрессионная модель. Авторегрессионная модель [110] порядка n использует параметрическую модель для аппроксимации временного ряда \mathbf{s} :

$$x_t = w_0 + \sum_{j=1}^{n-1} w_j x_{t-j} + \varepsilon_t,$$

где ε_t — регрессионные остатки. Оптимальные параметры \mathbf{w}^* авторегрессионной модели используются как признаки $\mathbf{g}(\mathbf{s})$. Данные параметры минимизируют квадратичную ошибку аппроксимации временного ряда и предсказания модели

$$\mathbf{x} = \mathbf{g}(\mathbf{s}) = \mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^n} \left(\sum_{t=n}^T \|x_t - \hat{x}_t\|^2 \right). \quad (6.3)$$

Задача (6.3) эквивалентна задаче линейной регрессии. Поэтому для каждого временного ряда s необходимо решить задачу линейной регрессии размера n . Пример аппроксимации временного ряда авторегрессионной моделью представлен на Рис. 6.1.

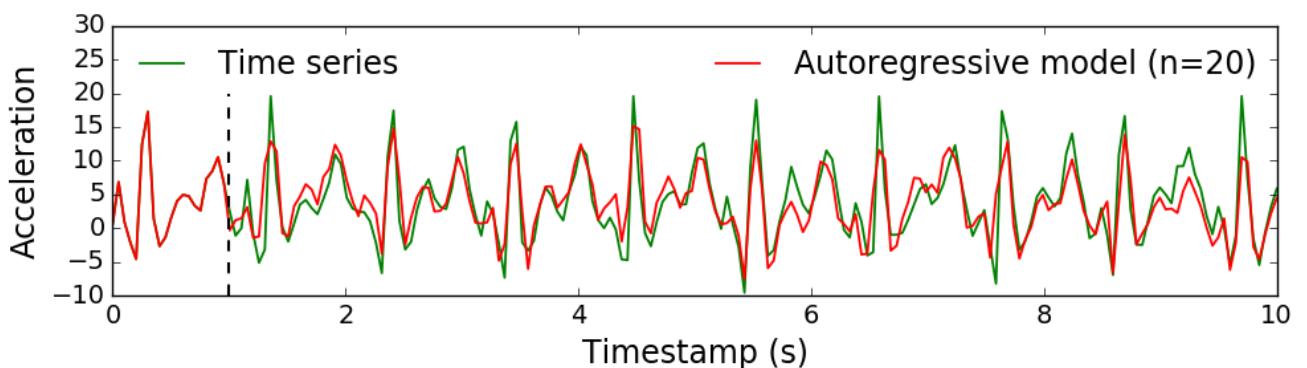


Рис. 6.1: Пример аппроксимации временного ряда авторегрессионной моделью с $n = 20$

Анализ сингулярного спектра. Для каждого временного ряда \mathbf{s} из исходной выборки строится траекторная матрица:

$$\mathbf{S} = \begin{pmatrix} s_1 & s_2 & \dots & s_n \\ s_2 & s_3 & \dots & s_{n+1} \\ \dots & \dots & \dots & \dots \\ s_{T-n+1} & s_{T-n+2} & \dots & s_T \end{pmatrix}.$$

Здесь ширина окна n является внешним структурным параметром. Сингулярное разложение матрицы $\mathbf{S}^T \mathbf{S}$:

$$\mathbf{S}^T \mathbf{S} = \mathbf{U} \Lambda \mathbf{U}^T,$$

где \mathbf{U} — унитарная матрица и $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ причём λ_i собственные значения $\mathbf{S}^T \mathbf{S}$. Признаковое описание временного ряда \mathbf{s} задаётся спектром матрицы $\mathbf{S}^T \mathbf{S}$:

$$\mathbf{x} = \mathbf{g}(\mathbf{s}) = [\lambda_1, \dots, \lambda_n]^T.$$

Аппроксимация сплайнами. Предлагаемый метод аппроксимирует временные ряды с помощью сплайнов [111]. Сплайн определяется его параметрами: узлами и коэффициентами. Предполагается, что узлы сплайна $\{\xi_\ell\}_{\ell=0}^M$ равномерно распределены по временной оси. Кусочные модели, построенные на отрезках $[\xi_{\ell-1}; \xi_\ell]$, заданы коэффициентами $\{\mathbf{w}_\ell\}_{\ell=1}^M$. Оптимальные параметры сплайна являются решением системы с дополнительными условиями равенства производных до второго порядка включительно на концах отрезков. Обозначим каждый отрезок-сегмент $p_i(t)$ $i = 1, \dots, M$ и весь сплайн $S(t)$. Тогда система уравнений принимает вид

$$S(t) = \begin{cases} p_1(t) = w_{10} + w_{11}t + w_{12}t^2 + w_{13}t^3, & t \in [\xi_0, \xi_1], \\ p_2(t) = w_{20} + w_{21}t + w_{22}t^2 + w_{23}t^3, & t \in [\xi_1, \xi_2], \\ \dots & \dots \\ p_M(t) = w_{M0} + w_{M1}t + w_{M2}t^2 + w_{M3}t^3, & t \in [\xi_{M-1}, \xi_M], \end{cases}$$

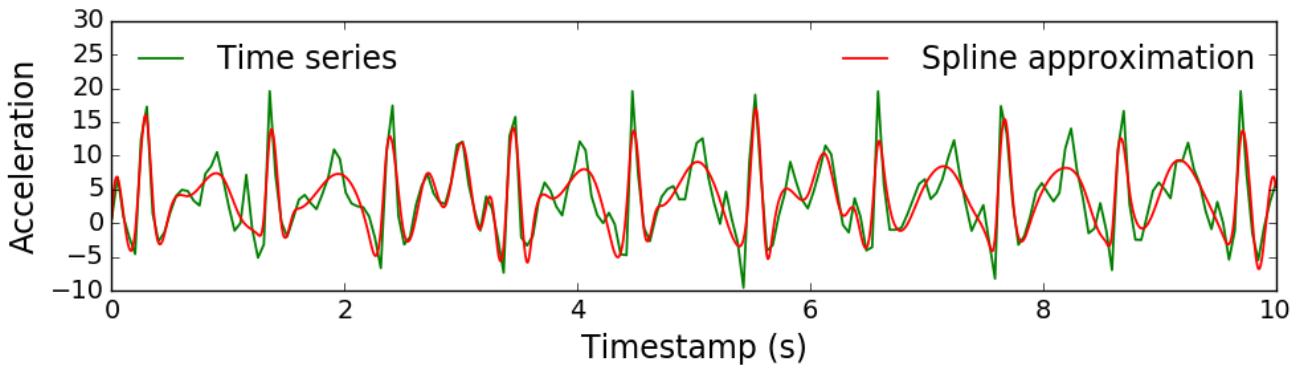


Рис. 6.2: Пример аппроксимации временного ряда с помощью сплайнов третьего порядка

$$\begin{aligned}
 S(\xi_t) &= x_t, \quad t = 0, \dots, M, \\
 p'_i(\xi_i) &= p'_{i+1}(\xi_i), p''_i(\xi_i) = p''_{i+1}(\xi_i), \quad i = 1, \dots, M-1, \\
 p_i(\xi_{i-1}) &= x_{i-1}, p_i(\xi_i) = x_i, \quad i = 1, \dots, M.
 \end{aligned}$$

Объединение всех параметров сплайна задаёт признаковое описание временного ряда:

$$\mathbf{x} = \mathbf{g}(\mathbf{s}) = [\mathbf{w}_1, \dots, \mathbf{w}_M]^T.$$

Рис. 6.2 показывает аппроксимацию временного ряда с использованием модели сплайнов. По сравнению с авторегрессионной моделью сплайны строят более гладкую аппроксимацию, используя такое же количество параметров.

6.3 Классификация временных рядов в порожденном признаковом пространстве

Для классификации временных рядов будем использовать подход один против всех. Для каждого класса обучается бинарный классификатор, и на стадии предсказания временной ряд классифицируется согласно наиболее уверенному классификатору. Использовались три модели классификации: логистическая регрессия, SVM и случайный лес.

Логистическая регрессия. Оптимальные параметры модели \mathbf{w}^*, b^* в случае логистической регрессии определяются минимизацией функции ошибки (6.1)

$$L(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}, \mu) = \sum_{i=1}^m \log(1 + \exp(-y_i[\mathbf{w}^\top \mathbf{x}_i + b])) + \frac{\mu}{2} \|\mathbf{w}\|^2, \text{ where } \boldsymbol{\theta} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}.$$

Решающее правило $f(\mathbf{x}, \boldsymbol{\theta})$ — знак линейной комбинации описания вектора \mathbf{x} и параметров $\boldsymbol{\theta}^*$

$$\hat{y} = f(\mathbf{x}, \boldsymbol{\theta}^*) = \operatorname{sgn}(\mathbf{x}^\top \mathbf{w}^* + b^*).$$

SVM. Оптимационная задача метода SVM имеет вид

$$\boldsymbol{\theta}^* = \begin{pmatrix} \mathbf{w}^* \\ b^* \\ \hat{\xi} \end{pmatrix} = \arg \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \mu \sum_{i=1}^m \xi_i, \text{ s.t. } y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad 1 \leq i \leq m.$$

Целевая функция соответствует функции ошибки классификации $L(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}, \mu)$. Предсказание для нового объекта вычисляется аналогично $\hat{y} = \operatorname{sgn}(\mathbf{x}^\top \mathbf{w}^* + b^*)$.

Случайный лес. Случайный лес использует идею бэггинга. Идея состоит в построении многих слабых, неустойчивых классификаторов на подвыборках с возвращениями и усреднения их предсказаний. Метод предполагает использование в качестве базовых классификаторов моделей с низким смещением и высокой дисперсией. Усреднение позволяет уменьшить дисперсию. В случае случайного леса базовой моделью выступают решающие деревья. Идея бэггинга используется не только для самих объектов, но и для множества признаков. В данном случае предсказание для нового объекта получается усреднением всех предсказаний отдельных деревьев:

$$\hat{y} = \text{sgn} \left(\frac{1}{B} \sum_{i=1}^B \text{pred}(\mathbf{x}_i) \right),$$

где B — количество деревьев в композиции.

6.4 Анализ порожденных признаковых пространств

В данной работе эксперименты проводились на двух наборах данных временных рядов акселерометра мобильного телефона: WISDM [106] и USC-HAD [112]. Акселерометр мобильного телефона проводит измерение ускорения по трём осям с частотой 100 Hz. Данные WISDM содержат 4321 временной ряд. Каждый временной ряд принадлежит к одному из 6 классов. Данные USC-HAD содержат 13620 временных рядов, принадлежащих одному из 12 классов. В таблице 6.2 представлено распределение временных рядов по классам для каждого датасета. Длина временного ряда равна 200. На Рис. 6.3 представлен пример одного из временных рядов.

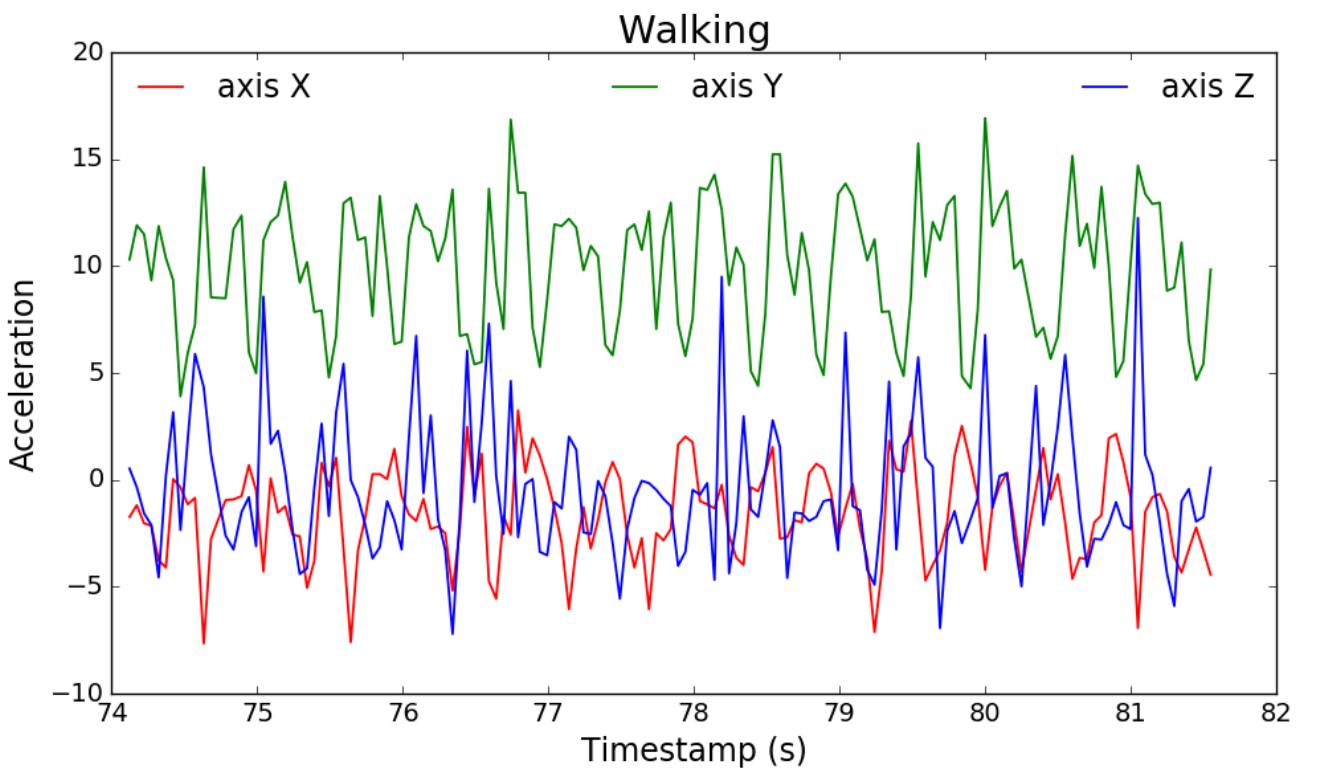


Рис. 6.3: Примеры временных рядов акселерометра для каждой оси

Таблица 6.2: Распределение объектов по классам для временных рядов акселерометра

(a) WISDM			(b) USC-HAD		
	Activity	# objects		Activity	# objects
1	Standing	229 5.30 %	1	Standing	1167 8.57 %
2	Walking	1917 44.36 %	2	Elevator-up	764 5.61 %
3	Upstairs	466 10.78 %	3	Walking-forward	1874 13.76 %
4	Sitting	277 6.41 %	4	Sitting	1294 9.50 %
5	Jogging	1075 24.88 %	5	Walking-downstairs	951 6.98 %
6	Downstairs	357 8.26 %	6	Sleeping	1860 13.66 %
Total		4321	7	Elevator-down	763 5.60 %
			8	Walking-upstairs	1018 7.47 %
			9	Jumping	495 3.63 %
			10	Walking-right	1305 9.58 %
			11	Walking-left	1280 9.40 %
			12	Running	849 6.23 %
			Total		13620

В эксперименте для каждого набора данных были порождены признаки одним из методов: экспертные функции, авторегрессионная модель, SSA и сплайны. Для каждой процедуры порождения признакового описания настраивались три модели классификации: логистическая регрессия, SVM и случайный лес. Внешние структурные параметры (длина авторегрессионной модели n , ширина окна SSA n , число узлов сплайна M) настраивались процедурой кросс-валидации:

$$CV(K) = \frac{1}{K} \sum_{k=1}^K \mathcal{L}(f_k, \mathbf{X}_k, \mathbf{y}_k, \boldsymbol{\mu}),$$

где $(\mathbf{X}_k, \mathbf{y}_k)$ — доля от всей выборки, используемая для обучения модели f_k .

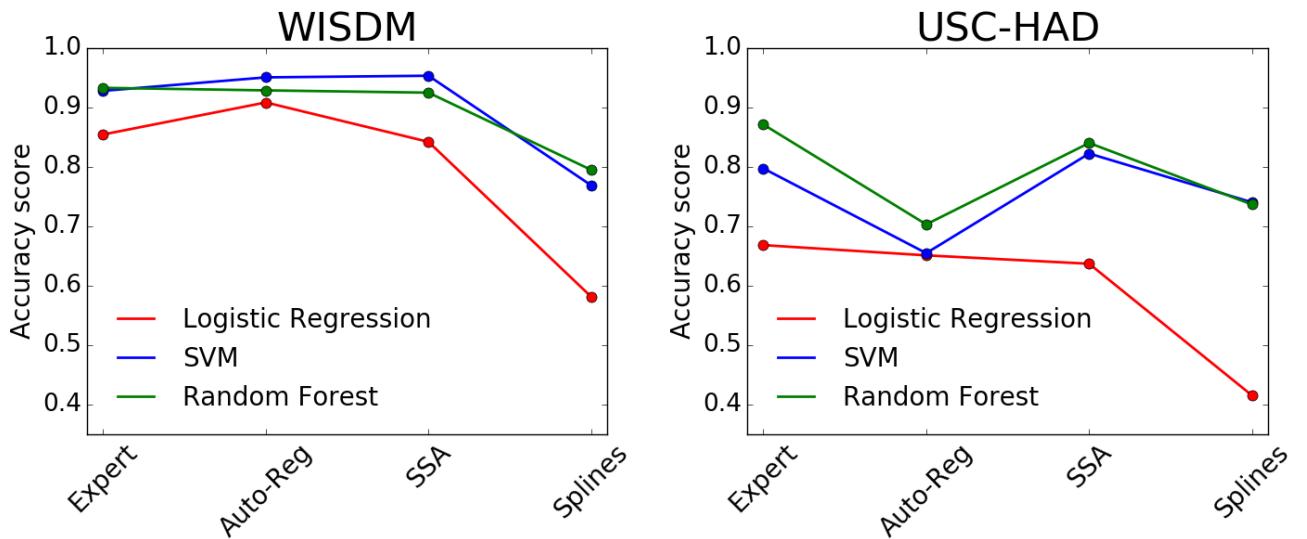


Рис. 6.4: Мультиклассовая точность классификации для различных порожденных признаковых пространств

Гиперпараметры μ моделей классификации были настроены той же процедурой кросс-валидации.

Первый подход к порождению признаков временных рядов — экспертные функции. Основной недостаток такого подхода необходимость экспертного задания функций и возможности их вычисления для конкретного набора данных.

Авторегрессионная модель требует задания параметра длины модели n . Процедура кросс-валидации дала наибольшее качество при $n = 20$ для обоих наборов данных.

Модель SSA была настроена аналогичной процедурой выбора оптимальных гиперпараметров. Конечная модель имела ширину окна $n = 20$.

Для аппроксимации временных рядов кубическими сплайнами [111] использовалась библиотека *scipy*. Узлы сплайнов $\{\xi_\ell\}_{\ell=1}^M$ были распределены равномерно по временной оси. Значение параметра M было подобрано на кросс-валидации.

Для обоих наборов данных процедуры порождения признаковых описаний дали следующие количества признаков: экспертные функции — 40; авторегрес-

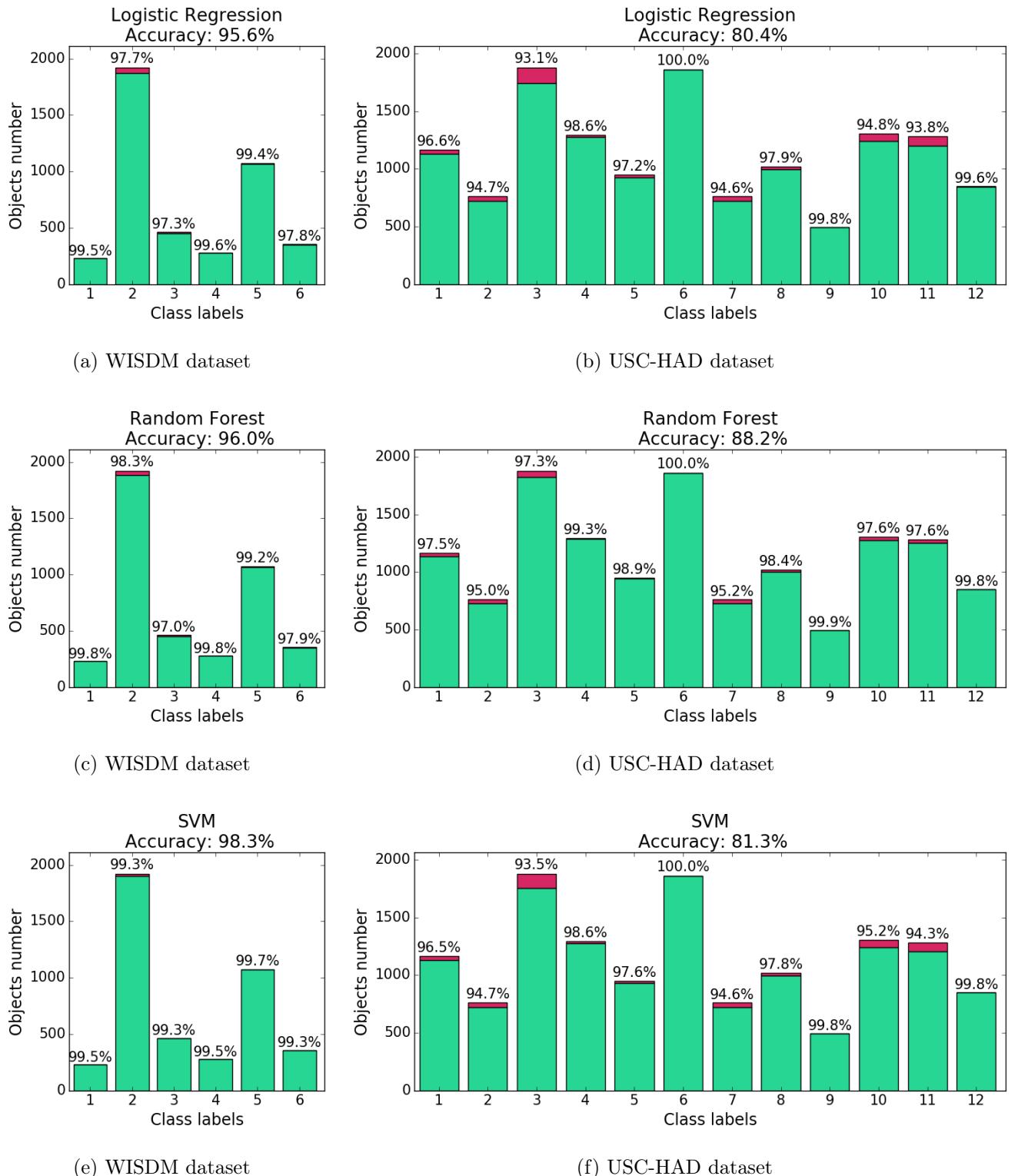


Рис. 6.5: Поклассовая точность классификации временных рядов акселерометра

сионная модель — 60; анализ сингулярного спектра — 60; сплайны — 33.

На Рис. 6.4 показано качество классификации (6.2) для двух наборов данных. Для данных WISDM сплайны дали самое слабое качество классификации.

Таблица 6.3: Бинарная точность классификации для данных WISDM с использованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				Random Forest				SVM			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.85	0.91	0.84	0.58	0.93	0.93	0.92	0.79	0.93	0.95	0.95	0.77
Standing	0.99	0.98	1.00	0.95	1.00	0.99	1.00	0.99	0.99	0.98	1.00	0.96
Walking	0.91	0.96	0.86	0.61	0.96	0.97	0.95	0.86	0.96	0.98	0.98	0.84
Upstairs	0.91	0.95	0.91	0.89	0.96	0.96	0.96	0.90	0.96	0.98	0.97	0.89
Sitting	0.99	0.98	1.00	0.99	1.00	0.99	1.00	1.00	0.99	0.98	1.00	1.00
Jogging	0.98	0.99	0.99	0.80	0.99	0.99	0.99	0.92	0.99	0.99	0.99	0.93
Downstairs	0.93	0.96	0.94	0.92	0.96	0.97	0.96	0.92	0.96	0.98	0.97	0.92

Таблица 6.4: Бинарная точность классификации для данных USC-HAD с использованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines

	Logistic Regression				Random Forest				SVM			
	EX	AR	SSA	SPL	EX	AR	SSA	SPL	EX	AR	SSA	SPL
All	0.67	0.65	0.64	0.41	0.87	0.70	0.84	0.74	0.80	0.65	0.82	0.74
Standing	0.94	0.94	0.92	0.89	0.98	0.94	0.97	0.98	0.95	0.94	0.97	0.96
Elevator-up	0.94	0.94	0.93	0.92	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-forward	0.87	0.87	0.89	0.70	0.97	0.89	0.96	0.88	0.95	0.87	0.97	0.91
Sitting	0.98	0.95	0.94	0.96	0.99	0.96	0.98	0.99	0.98	0.96	0.99	0.99
Walking-downstairs	0.95	0.93	0.93	0.90	0.99	0.96	0.98	0.95	0.98	0.93	0.98	0.96
Sleeping	1.00	0.98	0.99	1.00	1.00	0.98	1.00	1.00	1.00	0.98	1.00	1.00
Elevator-down	0.94	0.94	0.94	0.91	0.95	0.95	0.95	0.95	0.93	0.94	0.94	0.93
Walking-upstairs	0.94	0.95	0.93	0.92	0.98	0.95	0.98	0.96	0.98	0.95	0.98	0.96
Jumping	0.99	0.99	1.00	0.97	1.00	0.99	1.00	0.99	1.00	0.99	0.97	0.99
Walking-right	0.91	0.90	0.91	0.86	0.97	0.92	0.96	0.92	0.96	0.90	0.97	0.93
Walking-left	0.89	0.91	0.90	0.88	0.97	0.93	0.97	0.93	0.95	0.91	0.97	0.93
Running	0.99	0.99	0.99	0.92	1.00	0.99	1.00	0.97	1.00	1.00	0.95	0.98

Результаты для экспертных функций, авторегрессионной модели и SSA схожи. Для данных USC-HAD результат более восприимчив к выбору модели классификации. Для обоих наборов данных логистическая регрессия продемонстри-

ровала наименьшее качество, SVM и случайный лес показали почти одинаковое качество. Для набора данных USC-HAD модель с использованием аппроксимации сплайнами показала сравнимое с другими методами качество.

В таблицах 6.3 и 6.4 представлены результаты классификации (6.2) для каждого класса в отдельности. Первая строка в обеих таблицах демонстрирует точность по всем классам для каждой модели и процедуры генерации признаков. Следующие строки соответствуют бинарным точностям по каждому из классов. Для данных WISDM лучшее качество имеют наименее активные классы, такие как Standing и Sitting. Для USC-HAD заметного выделения качества для определенных классов не наблюдается.

Также был проведён эксперимент с использованием объединённого множества всех 193 сгенерированных признаков. Результаты представлены на Рис. 6.5. Соответствие между номера классов и видами активности приведено в таблице 6.2. Объединение признаков для обучения одной модели позволило увеличить качество. Для данных WISDM все точности классификации по классам больше 97%, а для USC-HAD выше 93%.

Заключение

Основные результаты диссертационной работы заключаются в следующем.

В главе 1 введено понятие прогностической модели в пространствах высокой размерности. Поставлена формальная задача декодирования сигналов. Приведён обзор методов снижения размерности сигналов. Рассмотрены линейные и нелинейные методы снижения размерности пространства. Описаны методы, работающие с тензорными и многомодальными данными.

В главе 2 введено понятие скрытого пространства и процедуры согласования. Рассмотрены методы построения согласованных моделей проекции в скрытое пространство. Доказано утверждение об оптимальности линейных методов проекции в скрытое пространство. Доказаны утверждения об оптимальности аддитивной суперпозиции моделей декодирования.

В главе 3 рассмотрена задача выбора признаков для декодирования сигналов. Задача выбора признаков ставится как задача дискретной оптимизации. Приведён метод выбора признаков с помощью квадратичного программирования как решение релаксированной оптимизационной задачи. Предложены обобщения процедуры выбора признаков для случая векторной целевой переменной: методы с симметричным и несимметричным учетом значимости целевых столбцов, а также минимаксная постановка задачи.

В главе 4 методы выбора признаков применяются к задаче выбора активных параметров при оптимизации нелинейных моделей. Предложена модификация метода Ньютона для повышения стабильности процедуры оптимизации. На каждом шаге алгоритма выбирается подмножество активных параметров для оптимизации алгоритмом выбора признаков с помощью квадратичного программирования.

В главе 5 ставится задача выбора оптимальной метрики в пространстве временных рядов. Приводится алгоритм кластеризации, использующий метрику Махalanобиса с обучаемой матрицей для вычисления расстояния между объ-

ектами. Для нахождения соответствия между временными рядами предложен алгоритм классификации с процедурой динамического выравнивания временных рядов.

В главе 6 рассмотрена задача порождения признакового пространства при решении задачи классификации временных рядов. Признаковое пространство порождается с помощью метамоделей временных рядов. В качестве метамоделей предлагаются авторегрессионная модель, метод анализа сингулярного спектра, аппроксимация сплайнами.

Список основных обозначений

\mathbb{X} — пространство исходной переменной

\mathbb{Y} — пространство целевой переменной

$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^{\top} = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]$ — исходная матрица

$\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^{\top} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r]$ — целевая матрица

$\mathbf{f}^{\text{AR}} : \mathbb{Y} \rightarrow \mathbb{Y}$ — авторегрессионная модель

$\mathbf{f}_{\text{R}} : \mathbb{X} \rightarrow \mathbb{Y}$ — регрессионная модель

Θ — параметры модели

$\mathcal{L}(\Theta, \mathbf{X}, \mathbf{Y})$ — функция ошибки прогностической модели

$\mathbf{s} = (s_1, \dots, s_T)$ — временной ряд длины T

$\mathcal{S} = \{\mathbf{s}^i\}_{i=1}^m$ — множество из m временных рядов

$\mathbf{x}_t = ([\mathbf{s}_{\mathbf{x}}^1]_t, \dots, [\mathbf{s}_{\mathbf{x}}^m]_t)$ — временное представление множества временных рядов

$\mathbf{X}_{t,h} = [\mathbf{x}_{t-h+1}, \dots, \mathbf{x}_t]^{\top}$ — представление предыстории

$\mathbf{Y}_{t,r} = [\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+p}]^{\top}$ — представление горизонта прогнозирования

\mathbb{T}, \mathbb{U} — скрытые пространства для пространств исходной и целевой переменных

$\varphi_{\mathbf{x}} : \mathbb{X} \rightarrow \mathbb{T}, \varphi_{\mathbf{y}} : \mathbb{Y} \rightarrow \mathbb{U}$ — функции кодирования

$\psi_{\mathbf{x}} : \mathbb{T} \rightarrow \mathbb{X}, \psi_{\mathbf{y}} : \mathbb{U} \rightarrow \mathbb{Y}$ — функции декодирования

$\mathbf{h} : \mathbb{T} \rightarrow \mathbb{U}$ — функция связи

$g : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ — функция согласования скрытых векторов

$S(\mathbf{a}', \mathbf{X}, \mathbf{Y})$ — функция ошибки выбора признаков

\mathbf{z} — вектор значимостей признаков

$\mathbf{Q}_x, \mathbf{Q}_y$ — матрицы парных взаимодействий исходных признаков и целевых столбцов

\mathbf{b} — вектор релевантностей признаков

$d_{\mathbf{A}} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ — расстояние Махаланобиса с матрицей трансформации \mathbf{A}

$G(\mathbf{x}, \{\mathbf{c}_e\}_{e=1}^K)$ — процедура выравнивания временных рядов относительно центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$

$\mathbf{g} : \mathbb{R}^T \rightarrow \mathbb{X}$ — функция порождения признаков

Список иллюстраций

1.1	Схема построения моделей декодирования	17
1.2	Модельный пример работы методов PCA и PLS	22
1.3	Модельный пример работы методов PCA и CCA	23
2.1	Сигналы мозга (левый график) и 3D координаты руки (правый график)	39
2.2	Прогноз потребления электроэнергии методом PLS при размерности латентного пространства $l=14$	40
2.3	Зависимость ошибки от размерности латентного пространства для данных потребления электроэнергии	41
2.4	Зависимость ошибки от размерности латентного пространства для данных ECoG	42
2.5	Прогноз движения руки по данным ECoG методом PLS при размерности латентного пространства $l = 5$	43
2.6	Зашумленные изображения из набора данных MNIST	44
2.7	Набор данных MNIST, в котором каждое изображение разделено пополам	44
2.8	Пример реконструкции правой части изображения по левой для рассматриваемых моделей	45
3.1	Значимости признаков \mathbf{z}_x и целевых столбцов \mathbf{z}_y в зависимости от α_3 для рассмотренного примера	51
3.2	Матрицы корреляций для матрицы плана \mathbf{X} и целевой матрицы \mathbf{Y} для данных ECoG	57
3.3	Значимости целевых столбцов \mathbf{z}_y в зависимости от α_3 для метода SymImp QPFS	58
3.4	Сравнение предложенных методов выбора признаков для данных ECoG при прогнозировании $k = 30$ отсчётов времени	60

3.5	Ошибка sRMSE на тестовой выборке для модели PLS	61
3.6	Диаграммы размаха значений sRMSE на тестовой выборке для моделей Lasso, Elastic, LinReg+QPFS, PLS, PLS+QPFS	61
4.1	Поверхность функции ошибки для логистической регрессии	69
4.2	Релевантность параметров для логистической регрессии	69
4.3	Поверхность функции ошибки для нейронной сети	70
4.4	Релевантность параметров первого слоя для модели нейронной сети	70
4.5	Оптимизационный процесс предложенного метода QPFS+Ньютон для модели логистической регрессии	71
4.6	Множество активных параметров на протяжении оптимизационного процесса	72
5.1	Сравнение оптимальной метрики Махаланобиса алгоритма LMNN с евклидовой метрикой в двумерном случае	81
5.2	Истинное распределение двумерных модельных данных	83
5.3	Результат кластеризации модельных данных алгоритмом k -средних	84
5.4	Результат кластеризации модельных данных алгоритмом адаптивного метрического обучения	85
5.5	Примеры центроидов синтетических временных рядов	86
5.6	Примеры центроидов временных рядов акселерометра	87
5.7	Примеры временных рядов акселерометра	87
5.8	Выравненные временные ряды акселерометра	88
5.9	Ошибка классификации метрического алгоритма в зависимости от размерности пространства и количества используемых ближайших соседей	88
6.1	Пример аппроксимации временного ряда авторегрессионной моделью с $n = 20$	93

6.2 Пример аппроксимации временного ряда с помощью сплайнов третьего порядка	95
6.3 Примеры временных рядов акселерометра для каждой оси	97
6.4 Мультиклассовая точность классификации для различных по- рожденных признаковых пространств	99
6.5 Поклассовая точность классификации временных рядов акселе- рометра	100

Список таблиц

1.1 Средняя квадратичная ошибка на модельном примере для методов линейной регрессии, PCA, PLS, CCA	23
2.1 Точность классификации линейного SVM для методов Deep CCA и CCA	42
2.2 Квадратичная ошибка для нелинейных моделей в задаче восстановления правой части изображения по левой	45
3.1 Обзор предлагаемых обобщений метода QPFS для векторной цепевой переменной	56
3.2 Стабильность предложенных методов выбора признаков	60
4.1 Средняя квадратичная ошибка рассматриваемых алгоритмов оптимизации для модели нелинейной регрессии	72
4.2 Среднее значение кросс-энтропии рассматриваемых алгоритмов оптимизации для модели логистической регрессии	73
5.1 Результаты кластеризации на множестве датасетов для методов k -средних и AML	84
5.2 Матрицы несоответствий для евклидовой метрики и метрики Махalanобиса, построенные для временных рядов акселерометра .	89
5.3 Прирост точности классификации при использовании адекватной оценки матрицы трансформаций	90
6.1 Примеры экспертных порождающих функций	92
6.2 Распределение объектов по классам для временных рядов акселерометра	98
6.3 Бинарная точность классификации для данных WISDM с использованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines	101

6.4 Бинарная точность классификации для данных USC-HAD с ис- пользованием рассматриваемых алгоритмов: EX — Expert, AR — Auto-Reg, SSA and SPL for Splines	101
---	-----

Литература

1. Anastasia Motrenko and Vadim Strijov. Multi-way feature selection for ecog-based brain-computer interface. *Expert Systems with Applications*, 114:402–413, 2018.
2. Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.
3. Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.
4. Hema Rao Madala and Alexey Ivakhnenko. *Inductive learning algorithms for complex systems modeling*. CRC press, 2019.
5. Alexandr Katrutsa and Vadim Strijov. Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications*, 76:1–11, 2017.
6. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.
7. Herman Wold. *Path models with latent variables: The NIPALS approach*. Elsevier, 1975.
8. Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *Proceedings of the 2005 international conference on Subspace, Latent Structure and Feature Selection*, pages 34–51. Springer, 2005.
9. Andrey Eliseyev, Vincent Auboironx, Thomas Costecalde, Lilia Langar, Guillaume Charvet, Corinne Mestais, Tetiana Aksanova, and Alim-Louis Benabid. Recursive exponentially weighted n-way partial least squares regression with recursive-validation of hyper-parameters in brain-computer interface applications. *Scientific reports*, 7(1):1–15, 2017.

10. Nina Golyandina, Vladimir Nekrutkin, and Anatoly A Zhigljavsky. *Analysis of time series structure: SSA and related techniques*. CRC press, 2001.
11. Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
12. Harold Hotelling. *Relations between two sets of variates*. Springer, 1992.
13. Alexandr Katrutsa and Vadim Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.
14. Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 11:1491–1516, 2010.
15. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer series in statistics New York, 2001.
16. Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.
17. Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.
18. Brian Kulis et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364, 2012.
19. Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Technical report, Michigan State Universiy, 2006.
20. Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2):207–244, 2009.

21. Jonathan R Wolpaw, Niels Birbaumer, William J Heetderks, Dennis J McFarland, P Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A Quatrano, Charles J Robinson, Theresa M Vaughan, et al. Brain-computer interface technology: a review of the first international meeting. *IEEE transactions on rehabilitation engineering*, 8(2):164–173, 2000.
22. Brendan Z Allison, Elizabeth Winter Wolpaw, and Jonathan R Wolpaw. Brain–computer interface systems: progress and prospects. *Expert review of medical devices*, 4(4):463–474, 2007.
23. Sebastian Nagel and Martin Spüler. Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain-computer interface. *PloS one*, 13(10):e0206107, 2018.
24. Xiang Zhang, Lina Yao, Xianzhi Wang, Jessica JM Monaghan, David Mcalpine, and Yu Zhang. A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers. *Journal of Neural Engineering*, 18(3), 2020.
25. Antonio Maria Chiarelli, Pierpaolo Croce, Arcangelo Merla, and Filippo Zappasodi. Deep learning for hybrid eeg-fnirs brain–computer interface: application to motor imagery classification. *Journal of Neural Engineering*, 15(3), 2018.
26. Andrey Eliseyev and Tetiana Aksenova. Recursive n-way partial least squares for brain-computer interface. *PloS one*, 8(7), 2013.
27. Andrey Eliseyev, Cecile Moro, Thomas Costecalde, Napoleon Torres, Sadok Gharbi, Corinne Mestais, Alim Louis Benabid, and Tatiana Aksenova. Iterative n-way partial least squares for a binary self-paced brain–computer interface in freely moving animals. *Journal of Neural Engineering*, 8(4), 2011.
28. Р. В. Исаченко. Метрическое обучение в задачах мультиклассовой классификации временных рядов. In *Ломоносов-2016*, pages 129–131, 2016.

29. R. G. Neychev, A. P. Motrenko, R. V. Isachenko, A. S. Inyakin, and V. V. Strijov. Multimodel forecasting multiscale time series in internet of things. In *Intelligent Data Processing*, pages 130–131, 2016.
30. Р. В. Исаченко, И. Н. Жариков, and А. М. Бочкарёв. Локальные модели для классификации объектов сложной структуры. In *Математические методы распознавания образов*, volume 18, pages 26–27, 2017.
31. R. V. Isachenko and V. V. Strijov. Dimensionality reduction for multicorrelated signal decoding with projections to latent space. In *Intelligent Data Processing*, pages 86–87, 2018.
32. Р. В. Исаченко and В. В. Стрижов. Снижение размерности в задаче декодирования временных рядов. In *Intelligent Data Processing*, pages 31–32, 2020.
33. Р. В. Исаченко and А. М. Катруца. Метрическое обучение и снижение размерности пространства в задачах кластеризации. *Машинное обучение и анализ данных*, 2(1):17–25, 2016.
34. Р. В. Исаченко and В. В. Стрижов. Метрическое обучение в задачах мультиклассовой классификации временных рядов. *Информатика и её приложения*, 10(2):48–57, 2016.
35. Roman Isachenko, Ilya Zharikov, Artem Bochkarev, and Vadim Strijov. Feature generation for physical activity classification. *Artificial Intelligence and Decision Making*, (3):20–27, 2018.
36. R. V. Isachenko and V. V. Strijov. Quadratic programming optimization with feature selection for nonlinear models. *Lobachevskii Journal of Mathematics*, 39(9):1179–1187, 2018.
37. R. V. Isachenko, M. V. Vladimirova, and V. V. Strijov. Dimensionality reduction for time series decoding and forecasting problems. *DEStech Transactions on Computer Science and Engineering*, (optim), 2018.

38. Ф. Р. Яушев, Р. В. Исаченко, and Б. В. Стрижов. Модели согласования скрытого пространства в задаче прогнозирования. *Системы и средства информатики*, 31(1), 2021.
39. George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.
40. Keith W Hipel and A Ian McLeod. *Time series modelling of water resources and environmental systems*. Elsevier, 1994.
41. John H Cochrane. Time series for macroeconomics and finance. *Manuscript, University of Chicago*, pages 1–136, 2005.
42. John W Galbraith, Victoria Zinde-Walsh, et al. Autoregression-based estimators for arfima models. Technical report, CIRANO, 2001.
43. Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.
44. Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.
45. Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, III. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.
46. Herman Wold and Jean-Luc Bertholet. The pls (partial least squares) approach to multidimensional contingency tables. *Metron*, 40(1-2):303–326, 1982.
47. Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.
48. Paul Geladi. Notes on the history and nature of partial least squares (pls) modelling. *Journal of Chemometrics*, 2(4):231–246, 1988.

49. Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.
50. V Esposito Vinzi, Wynne W Chin, Jörg Henseler, Huiwen Wang, et al. *Handbook of partial least squares*, volume 201. Springer, 2010.
51. Richard G Brereton and Gavin R Lloyd. Partial least squares discriminant analysis: taking the magic away. *Journal of Chemometrics*, 28(4):213–225, 2014.
52. Roman Rosipal. Nonlinear partial least squares: an overview. *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*, pages 169–189, 2011.
53. Theodore Wilbur Anderson. An introduction to multivariate statistical analysis. Technical report, Wiley New York, 1962.
54. Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.
55. Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks*, pages 353–360. Springer, 2001.
56. Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48, 2002.
57. David R Hardoon, Sandor Szegedy, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.
58. David R Hardoon, Janaina Mourao-Miranda, Michael Brammer, and John Shawe-Taylor. Unsupervised analysis of fmri data using kernel canonical correlation. *NeuroImage*, 37(4):1250–1259, 2007.
59. Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances in neural information processing systems*, pages 1497–1504, 2002.

60. Luca Montanarella, Maria Rosa Bassani, and Olivier Bréas. Chemometric classification of some european wines using pyrolysis mass spectrometry. *Rapid Communications in Mass Spectrometry*, 9(15):1589–1593, 1995.
61. Jean-Philippe Vert and Minoru Kanehisa. Graph-driven feature extraction from microarray data using diffusion kernels and kernel cca. In *Advances in neural information processing systems*, pages 1449–1405, 2003.
62. Aria Haghghi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: Hlt*, pages 771–779, 2008.
63. Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-view learning of word embeddings via cca. In *Advances in neural information processing systems*, pages 199–207, 2011.
64. K Choukri and G Chollet. Adaptation of automatic speech recognizers to new speakers using canonical correlation analysis techniques. *Computer Speech & Language*, 1(2):95–107, 1986.
65. Frank Rudzicz. Adaptive kernel canonical correlation analysis for estimation of task dynamics from acoustics. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4198–4201, 2010.
66. Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
67. Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.
68. Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092, 2015.

69. Xiaobin Chang, Tao Xiang, and Timothy M Hospedales. Scalable and effective deep cca via soft decorrelation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1488–1497, 2018.
70. Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
71. Qibin Zhao, Cesar F Caiafa, Danilo P Mandic, Zenas C Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang, and Andrzej Cichocki. Higher order partial least squares (hopls): a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1660–1673, 2012.
72. Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5), 2016.
73. Einat Kidron, Yoav Y Schechner, and Michael Elad. Pixels that sound. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 88–95. IEEE, 2005.
74. Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136, 2009.
75. Raman Arora and Karen Livescu. Kernel cca for multi-view learning of acoustic features using articulatory measurements. In *Symposium on Machine Learning in Speech and Language Processing*, 2012.
76. Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 966–973. IEEE, 2010.

77. Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.
78. Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in neural information processing systems*, pages 1853–1861, 2014.
79. Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.
80. Jonathan Masci, Michael M Bronstein, Alexander M Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):824–830, 2013.
81. Janarthanan Rajendran, Mitesh M Khapra, Sarath Chandar, and Balaraman Ravindran. Bridge correlational neural networks for multilingual multimodal representation learning. *arXiv preprint arXiv:1510.03519*, 2015.
82. Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421, 2011.
83. Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167. IEEE, 2012.
84. Paul Geladi. Notes on the history and nature of partial least squares (pls) modelling. *Journal of Chemometrics*, 2(January):231–246, 1988.
85. Ragnar Frisch and Frederick V Waugh. Partial time regressions as compared with individual trends. *Econometrica: Journal of the Econometric Society*, 1(4):387–401, 1933.

86. Michael C Lovell. Seasonal adjustment of economic time series and multiple regression analysis. *Journal of the American Statistical Association*, 58(304):993–1010, 1963.
87. Carlos Cinelli and Chad Hazlett. Making sense of sensitivity: Extending omitted variable bias. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(1):39–67, 2020.
88. Kentaro Shimoda, Yasuo Nagasaka, Zenas C Chao, and Naotaka Fujii. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in Japanese macaques. *Journal of neural engineering*, 9(3):036015, 2012.
89. Zenas C Chao, Yasuo Nagasaka, and Naotaka Fujii. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in neuroengineering*, 3:3, 2010.
90. Yann LeCun, Corinna Cortes, and Chris Burges. The mnist dataset of handwritten digits. Available at: <http://yann.lecun.com/exdb/mnist/index.html>, 1998.
91. Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
92. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
93. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
94. Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016.
95. Mordecai Avriel. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

96. Barbara Blaschke, Andreas Neubauer, and Otmar Scherzer. On convergence rates for the iteratively regularized gauss-newton method. *IMA Journal of Numerical Analysis*, 17(3):421–436, 1997.
97. Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In *International Conference on Machine Learning*, pages 557–565, 2017.
98. Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.
99. Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.
100. Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository. Available at: <http://archive.ics.uci.edu/ml>, 2017.
101. Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.
102. John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
103. Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. *KDD workshop*, 10(16):359–370, 1994.
104. Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
105. Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
106. Gary M Weiss. The wisdm: Wireless sensor data mining dataset. Available at: <http://www.cis.fordham.edu/wisdsm/dataset.php>, 2013.

107. M.E. Karasikov and V.V. Strijov. Feature-based time-series classification. *Intelligence*, 24(1):164–181, 2016.
108. M.P. Kuznetsov and N.P. Ivkin. Time series classification algorithm using combined feature description. *Machine Learning and Data Analysis*, 1(11):1471–1483, 2015.
109. Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
110. Yu P Lukashin. Adaptive methods of short-term forecasting of time series. *M.: Finance and statistics*, 2003.
111. Carl De Boor. *A practical guide to splines*. Springer-Verlag, 1978.
112. Mi Zhang and Alexander A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. *Available at: <http://sipi.usc.edu/had/>*, 2012.